ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 2, No. 3, 553-561 (2017)
www.astesj.com
Special Issue on Recent Advances in Engineering Systems

ASTES Journal
ISSN: 2415-6698

Semantic-less Breach Detection of Polymorphic Malware in Federated Cloud

Yahav Biran[*,1], George Collins[1], Borky John M[1], Joel Dubow[2]

[1]*Colorado State University, Department of Systems Engineering, Fort Collins, 80523, CO*

[2]*Fulcrum Co., Cyber Security, Centreville, 20120, VA*

ARTICLEINFO

Article history:
Received: 10 April, 2017
Accepted: 20 May, 2017
Online: 01 June, 2017

Keywords:
Anomaly detection
Polymorphic Malware
Cloud Federation

ABSTRACT

Cloud computing is one of the largest emerging utility services that is expected to grow enormously over the next decade. Many organizations are moving into hybrid cloud/hosted computing models. Single cloud service provider introduces cost and environmental challenges. Also, multi-cloud solution implemented by the Cloud tenant is suboptimal as it requires expensive adaptation costs. Cloud Federation is a useful structure for aggregating cloud based services under a single umbrella to share resources and responsibilities for the benefit of the member cloud service providers. An efficient security model is crucial for successful cloud business. However, with the advent of large scale and multi-tenant environments, the traditional perimeter boundaries along with traditional security practices are changing. Defining and securing asset and enclave boundaries is more challenging, and system perimeter boundaries are more susceptible to breach. This paper to describe security best practices for Cloud Federation. The paper also describes a tool and technique for detecting anomalous behavior in resource usage across the federation participants. This is a particularly serious issue because of the possibility of an attacker potentially gaining access to more than one CSP federation member. Specifically, this technique is developed for Cloud Federations since they have to deal with heterogeneous multi-platform environments with a diverse mixture of data and security log schema, and it has to do this in real time. A Semantic-less Breach detection system that implements a self-learning system was prototyped and resulted in up to 87% True-Positive rate with 93% True-Negative.

1 Introduction

As an increasing fraction of computing services move to the Cloud, there will be a proliferation of software characteristics, service models, and deployment options. Many organizations are moving into hybrid cloud/hosted computing models. A Cloud Service Provider (CSP) goal is to maximize its market share among the potential Service Providers (SP). Accommodating variable demands for computing resources requires an immense capacity, as it calls for providing for the maximum demand. In some cases, this drives them to underutilization of massive datacenter deployments. In other situations, the CSPs suffer over-utilization because of a miss in the market share, load and reliability projections. Both cases lead to sub-optimal utilization.

From the SPs perspective, they are most interested in availability and adaptability. The former refers to reliable service conditions that make its services available to the users it serves. The latter relates to the Vendor lock-in risk[1]. Single CSP provides a sub-optimal solution to the SP thus multi-cloud become an attractive solution. However, multi-cloud solution implemented by the SP requires expensive adaptations to the CSP's tools and service constructs that may vary among different CSPs.

Cloud Federation is a new paradigm that allows many CSPs to utilize computing resources optimally[2, 3]. Also, it allows SP to avoid the Vendor lock-in risk and provide service availability that can not be provided by a single CSP. No matter what the architecture, there is a need for to ensure the security and information assurance to users. Cloud Federation is an advantageous structure for aggregating cloud based services under a single umbrella to share resources and responsibilities for the benefit of the member cloud ser-

[*]Corresponding Author, ybiran@colostate.edu

https://dx.doi.org/10.25046/aj020371

vice providers. Federation is useful not only for sharing resources amongst cloud service providers but also for providing enclaves for interactions to perform domain-specific missions such as electrical grids and supply chains.

The Federation will need to assure that data transfers amongst the Federation's CSPs are secure. The Federation will, above all, need to detect any anomalous behavior occurring in transactions and resource sharing. In addition to the growing number of security tools, there is a need to log and identify security issues requiring attention early on in the process. In particular, breach detection in inter-cloud data transfer and communications is a particularly serious issue because of the possibility of an attacker potentially gaining access to more than one CSP federation member.

This paper to describe security best practices for Cloud Federation. The paper also describes a tool and technique for detecting anomalous behavior in resource usage across the federation participants. Specifically, this method is developed for Cloud Federations since they have to deal with a heterogeneous multi-platform environment with a diverse mixture of data and security log schema, and it has to do this in real time. This Semantic-less tool is described below after a description of the context of the issue.

The reminder of paper is organized as follows, Section II (Federated Cloud) discusses a new inter-cloud structure. It also describe some of the terminology used in this paper; Section III (Cyber Security Challenges in Federated Cloud) describes the core challenges of such inter-cloud system in a multi-layer model; Section IV (Semantic-less Breach Detection) discuss the tool we suggest for detecting the behavior of the anomalous system that runs in the Cloud Federation; Section V (Evaluation) discuss the breach-detection tool prototyped; Section VI (Analysis) analyze and present the prototype results; finally, Section VII (Conclusions) summarizes the main results and ongoing developments.

2 Federated Cloud

Organizations that serve and process a large number of simultaneous users and the user's often large quantities of data are termed "cloud computing services." These services enable convenient, on-demand, network access to a shared pool of configurable computing resources. The following section describes cloud federation and the reasons for its formation. It serves as background for the security design and breach detection mechanism we propose in a later section.

Cloud computing is occupying a rapidly growing market share of IT resources. IT related spending toward workload processing increased 32.8 percent in 2014, 29 percent in 2015, 34.1 percent during 2016, and expected to grow 42 percent during 2017[1]. Also, from environmental perspectives, grid energy powered by hydrocarbons are increasingly devoted to the growing power needs of cloud computing data centers. Carbon emission generated by data centers is growing from a 2011 level of 21.3 MtCO2e and expected to rise to 39.1 MtCO2e by 2020[2].

This makes cloud computing one of the fastest growing consumers' utility services, one that is projected to grow an order of magnitude in the next decade. [2] propose a new paradigm that will allow multiple cloud providers to utilize computing and energy resources optimally. From the customer's perspectives, the organization's total-cost-of-ownership is expected to shift from capital-expense-based, e.g., on-premise deployments to operational-expense-based, e.g., cloud service subscriptions. Thus, there is a need to create an efficient and transparent eco-system that allows the organization to match IT expenses with its planned cost structure.

Federated Clouds demonstrate a new paradigm that will allow multiple cloud providers to utilize optimally computing resources. It will do this by (1) lowering the data center's deployments per provider ratio, share and (2) scheduling available energy via aggregators and (3) lastly to employ, where appropriate, more renewable and carbon-free energies. It will quantify results and baseline the efforts based on work at the data centers within a single cloud provider. The federated cloud demonstrates the utility software container-based paradigm for achieving dense and elastic containerization technologies. It is centered on several discreet Linux Containers (LXC) managed by a Kubernetes resource management system [3] that acts as the governance engine. Moreover, the proposed solution scales out and optimizes cross-data centers and cross-regional deployments by computing and suggesting a cross-cloud provider's collaboration via a cloud aggregator. Furthermore, it suggests operating data centers employing maximum intermittent green energy sources[2]. Yet all of these advantages will be for naught if federation services cannot be supplied securely in the face of growing sophistication and quantities of cyber security threats. It is therefore essential that risks to data, computing resources and communications are managed such that the value of services provided exceeds the losses arising from cyber breaches.

2.1 Architecture

The Cloud Federation architecture is comprised of multiple CSPs, a Clouds-Coordinator, and a Clouds-Broker system. These are defined below.

Clouds-Coordinator. Acts as an information registry that stores the CSPs pricing offers and demand patterns. Clouds Coordinators periodically update the CSPs availability and offering prices. Also, a Clouds-Coordinator will help to employ, where appropriate, more renewable and carbon-free energies [3].

Clouds-Broker. Manages the membership of the CSP constituents. Both CSPs and SPs will use the Clouds-Broker to onboard the Cloud Federation. Also, Clouds-Broker will acts on behalf of the SP for resource allocation and provisioning requests. Clouds-Broker also provides a continuous ability to deploy SP software, configuration, and data to one or more CSP, so it provides the SoS IT agility goal.

[1] Gartner Says Worldwide Cloud Infrastructure-as-a-Service Spending to Grow 32.8 percent in 2015

[2] GeSI SMARTer2020 The Role of ICT in Driving a Sustainable Future. 2015. GeSI

[3] http://kubernetes.io

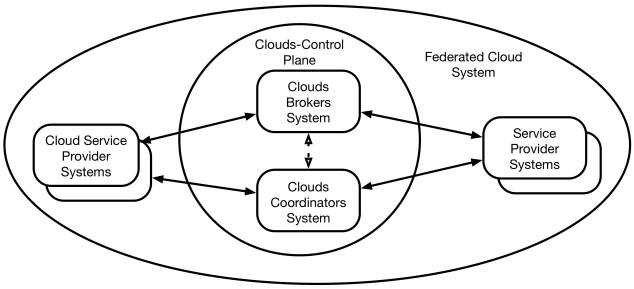

Figure 1: Proposed Cloud Federation Systems of System comprises of CSPs and SP that are managed by Clouds-Broker and Clouds-Coordinator

3 Cyber Security Challenges in Federated Cloud

The Cloud Federation has a global scale software and hardware infrastructure. We describe a progressive layers security model starting from the physical security of data centers, progressing to the hardware and software that underlies the infrastructure, and the constraints and processes to support the Cloud Federation operational security. The following section describes the Cloud Federation cyber security design throughout the data processing life cycle at a Cloud Federation. e.g., enables secure communication with tenants (SP) and its customers or control plane communication including CSP, Cloud-Brokers, and Clouds-Coordinator.

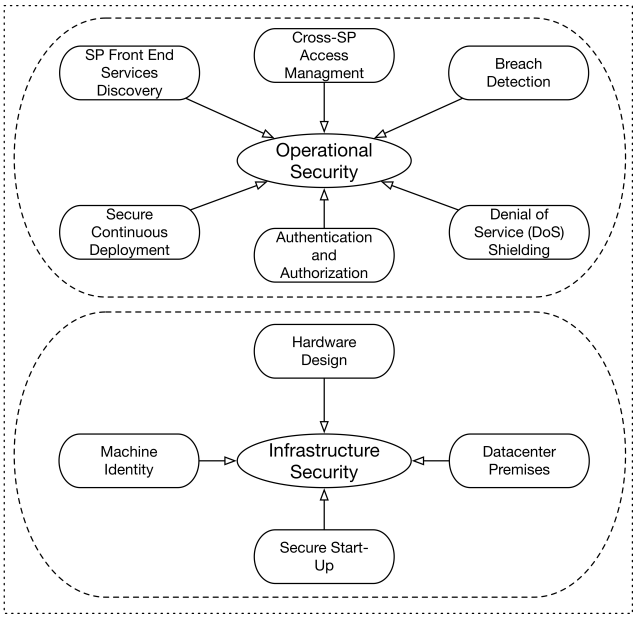

Figure 2: Cloud Federation Cyber Security Model includes two core layers, infrastructure security and operational security

Figure 2 describes the cyber security layers offered by the Cloud Federation. The following paragraph briefly describes the security elements corresponded with each layer[4]. Our extended cyber security model will emphasize the operational security with unique breach detection methodology. This was done since the operational security corresponds to the perimeter security of an enterprise system and the interface to the Federation members. Also, it will suggest a system for encryption of both inter microservices communication with emphasis on cross-CSP for tenants' workloads.

3.1 Infrastructure Security

The required baseline security level needed for cloud federation constituent's systems is referenced in Figure 1. It includes deployed facilities and computer systems managed by the CSPs or the Federation. The larger CSP's often exceed these baselines.

Datacenter Premises. CSPs design and build its data centers based on its expected computing capacities and service reliability manifested by their SLA and the redundancy levels of sub systems[4, 5]. The datacenter incorporates various components of physical security protections. Access to such facilities is governed by the CSP security operations. It uses technologies such as biometric identification, metal detection, metal detectors, and CCTV solutions[6].

Hardware Design. CSPs data centers run computing server machines fed by power distribution units and connected to a local network that is all connected to the edge of the wide network. The computing, digital storage, and networking equipment require a standard that ensures the required audit and validation of the security properties by components[7, 8], e.g., hardware security chip [9].

Machine Identity. confirms that any participating computing server in Cloud Federation can be authenticated to its CSP machine pool throughout a low-level management services[9]
Secure Start-Up. Ensures that CSPs servers are booting the correct software stack. Securing underlining components such as Linux boot loaders, OS system images and BIOS by cryptographic signatures can prevent an already compromised server from being continuously compromised by an ephemeral malware.

3.2 Operational Security

Operational security comprises the business flows between the SP with the cloud federation and the CSP it uses for processing workloads. The following section briefly discusses the required cybersecurity measures needed for SP and CSP business scenarios in a cloud federation.

Cross-SP Access Management: SP workloads are manifest in two workload types, (1) short-lived workloads. i.e., jobs that are terminate upon completion, and (2) long-lived workloads. i.e. services. The former workload might require connectivity to external services during its processing. The latter might expose serving endpoints to other services. e.g., short-lived jobs might require persistent storage to write its job results hence connecting to BigTable[5] storage server provisioned by other CSPs, which, in turn, require access management that uses credentials and certificates

[4]We extrapolate Google Cloud security model from https://cloud.google.com/security/security-design/
[5]https://cloud.google.com/bigtable/

stored within the Cloud Federation.

SP Front End Service Discovery: Long-lived workloads might expose public facing endpoints for serving other workloads or end-user requests. SP front-end services require publishing endpoints to allow other workloads within or external to the cloud Federation to discover their public facing entry point and this requires service discovery capabilities. Service discovery endpoints, and the actual service endpoints, are prone to risks such as Denial of Service attacks or intrusions originated by an attacker. We argue that current solutions offered by individual CSP's are sub-optimal because of the target scope of the intrusion. i.e., assuming an attack probability for a given CSP, running several CSPs reduces the risk by a factor of the number of CSPs. Later sections will formulate the risk function and show how cloud-federation minimizes those challenges by using the semantic-less breach detection system and show how most risks originate by crossing machine boundaries.

Secure Continues Deployment: Continuous Deployment (CD) is the function that allows cloud-native applications to get updated through an automated pipeline that is initiated by a new or updated code submission, compiled, tested through various quality gates until it is certified for deployment of the production systems and deployed seamlessly. Continues deployment enables cloud applications to innovate faster and safer no matter what number of machines are in the service pool. A secure continuous deployment service requires secured SP code and a configuration repository that authenticates to the target computing resource regardless of the CSP network segmentation. Traditional network segmentation, or fire walling, is a secondary security mechanism that is used for ingress and egress filtering at various points in the local network segment to prevent IP spoofing[10, 11]

Authentication and Authorization. In a federated cloud architecture, deployed workloads might require access to other services deployed by the federation. The canonical example will be an end user request service deployed in the Federation that triggers another micro-service within the SP architecture. Such cascading requests require multilayered authentication and authorization processes. i.e., a micro-service calls another micro-service and authenticate on behalf of the end user for audit trails supported by the end-user authentication token and the cascading micro-service tokens generated throughout the end user request. Figure 3 depicts the data flow during a call initiated by SP Micro-Service that runs in one of the federation's CSP denoted by CSP_i and CSP_j. A call initiated from SP_n that was provisioned in the federation as ms_n. The call destination runs on a different and sometimes the same SP. Let SP_m denote the destination SP. The call payload is encrypted by SP_n private key. The call arrived at an SP_m endpoint and checked for admission. SP_m admission control decrypts the call payload using SP_n public key that was submitted throughout the on-boarding process to the Cloud Federation. It is verified for authenticity and authorization of allowed call-sets. If admitted, SP_m calls and process the get_data() call and sends back the response to the originating SP, SP_n.

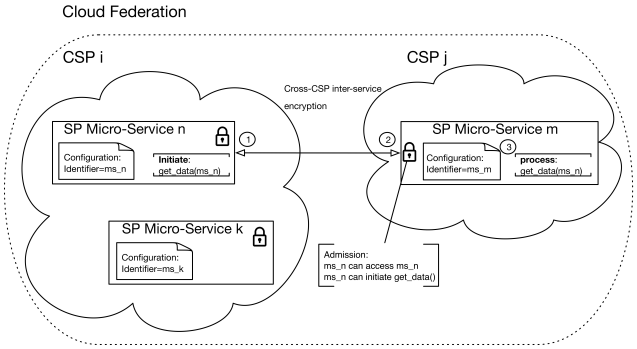

Figure 3: Authentication and authorization in Cloud Federation Cross-SP model

Breach Detection: The Cloud federation comprises various workload types that are owned by different autonomous organizations. Breach detection includes a complex data processing pipeline that integrates system signals originated from specific users of a CSP service as well as the potential cloud federation tenants. System signals are comprised of network devices as well as signals from infrastructure services. Only in recent years, after the growing numbers of data breaches and liabilities arising from losses,[12, 13, 14] have organizations started to incorporate business related metrics for breach detection[15]. Both data pipelines need to generate operational security warnings of potential incidents. The output of such warnings usually alerts security operations staff to potential incidents that require the relevant team's triage and response as appropriate.

Such methods are sub-optimal in a Federated Cloud for two main reasons: (1) different data sets are owned by different organization departments that are not integrated physically, schematically or semantically, (2) Lack of unification of both data sets as accomplished by fusion requires a complex transformation of both data sets semantics into a single data set. The above situation exacerbated when migrating the workload to the cloud as it introduces another orthogonal data set that contributes to complexity. The following sections propose a method for breach detection that collapses the three silos into a cohesive semantic-less data set that will enhance the Cloud Federation services detection breaches to an extent limited by available data and their investment in detection .i.e. allowing methods to the tenants to incorporate more data about their workload for more automatic detection.

4 Semantic-less Breach Detection

Malware infected cloud-computing-workloads introduces three core risks for organizations (1) Service unavailability, (2) Data breach , and (3) Data corruption. There is a need for breach detection system that helps to determine whether a workload is infected as well as the type of exploited risk type as enumerated above. Breach detection system effectiveness is influenced by a number of factors. We focused on the human social factor and the emergent public cloud offering. The following paragraph describes the important

factors required for optimized breach detection. This mode of breach detection has to span the heterogeneous schema employed by the various federation members.

4.1 The Human Social Factor

Enterprise IT is typically organized into silos. e.g., IT operations, network operations, database administration, and product engineering. The silos goal is to allow field-based ownership. Usually, silo teams are governed by different management hierarchies, communication styles, and vocabularies i.e. *semantics*. As as far as cyber security goes, semantics manifest by a particular interpretation of intrusion or breach. e.g., malware sending data to C&C might not impact the normal operation of a workload. Thus, product owners are oblivious to that risk while network operations detect unusual egress or ingress traffic usage patterns.

Enterprise IT workloads deployed as SP workload requires adopting unified cyber security best practices that overcome the different management hierarchies, communication styles, system security pans, data scheme, semantics and, vocabularies. The next paragraph shows how Federated Cloud helps enterprise IT improve its cyber security resiliency by offering a prediction tool that allows SP to apply proactive policies to mitigate potential threats.

4.2 The Cloud Federation Factor

Public Cloud services exacerbate the organization's human factor risk by introducing an additional silo that is often separated from the organization it serves. Public cloud operations are agnostic to its tenant's workload semantics by definition. CSPs configure their multi-tenancy to allow business with conflict of interest to run its workload on the same platform. Such practices and policies, augment the lack of cohesive view required for optimized malware detection.

Workloads deployed in public cloud services are not limited to known machine boundaries as traditional on-premise models offer. Although CSPs feature cyber security mechanisms that attempt mimicking the traditional computing workload hosting, workloads artifacts are under the CSP control. As such, the cloud client workloads might be compromised. Thus, there is a need for another cyber security dimension for the SP workload that overcomes the lack of control when running in the cloud.

We proposed a unique self-learning methodology that removes the need for tenant information that streamlines semantic-less information from the various software stacks of the Cloud Federation, including both tenant metrics and control-plane metrics. Also, it streamlines training data of security incidents shared in collaborative platforms outside the Cloud Federation. We also argue that a Cloud Federation optimizes such collaboration and self-learning process. We prototyped a system that implements such self-learning system that resulted in up to 87% True-Positive rate with 93% True-Negative.

Workload data and usage patterns form a critical path for the SP business success. The leakage of some of the workload data and usage patterns impose a threat to the SP business. This challenge represents a new threat of organizational espionage as well as attacks on the SP service that impacts SP business continuity. Therefore, sharing semantics breaks the isolation between the two systems and might hold the hosting system accountable for security attacks in CSP or Cloud Federation platforms. Also, transforming every workload semantics into a coherent model that aggregates numerous SP workloads requires a significant amount of investment. SPs will be reluctant to make such an investment, especially since it doesn't produce income. Therefore, this method has a low likelihood of being implemented. Therefore, enabling a method that eliminates SP investment and business risks is a key for the breach detection system success. Finally, a Cloud-Federation provides a centralized view of cross-CSP operations. Such centralized view allows SP workload deployment to different CSPs to gather a rich data set that will be available for malware identification and later, for predictive analytics. We suggest a method that captures computing resources usage and intra federation traffic and infers potential breach or disruption to proactively alerts CSP security stakeholders about suspicious cyber instances.

4.2.1 From Workload Semantics to Semantic-less

Cloud workloads are broadly composed of two types: online system, and offline system. The former provides low-latency, read/write access to data. For example, a web user requests a web page to load online and serve within a fraction of a second. The latter provides batch-like computing tasks that process the data offline, which is reported later to users by the system servers; for example, the search results based on a pre-calculated index. Offline production workloads are usually comprised of mainly unstructured data sets, such as click stream, web graph, and sensors data[2, 3].

The semantic-less detection will address the polymorphic malware case as its data stream are abstracted from computing activity. More specifically, a tenant's workload in a federated cloud manifested by software containers that are limited to not more than (1) namespace per tenant for isolation and (2) limited to a resources control groups(aka cgroup)[6] Control groups are the mechanism for limiting computing server host CPU, Memory, Disk I/O and Network I/O usage per namespace. That is the foundation of Linux Containers, which alludes to the existing methods of measurements of the metrics set, CPU, memory and I/O usage. We call this set the behavioral attributes set. Access to cgroup and namespace configuration and control is available on the host level i.e. the host OS that runs the multi-tenant workloads i.e. a control-plane component.

4.3 Data Collection

Both Cyber Security leaders and national agencies agree that addressing emerging cyber risks require sharing cyber attacks retrospects and their historical behavior, and discovered vulnerability reports as a foundation for collaboration, predictive time series analysis, risk quantification and risk allocations all leading to safer cyber services [16, 17]. Incidents are often documented in unstructured reports that

[6] https://www.kernel.org/doc/Documentation/cgroup-v1/cgroups.txt

require a manual analysis to identify trends[18].

To assess whether or not a system was breached, it is required to establish malicious system behavior patterns and then decompose those patterns into generic computing system metrics that can later be classified as harmful or safe. The following paragraph includes the source datasets we chose to assess the initial malicious patterns and their detectability by our method. We continued by decomposing the data and removing the tenant semantics. That allows a generic pattern of malicious activity dataset that can be used as a training data for the supervised model.

4.3.1 Source Datasets

We choose the National Vulnerability Database (NVD)[19] and the Vocabulary for Event Recording and Incident Sharing (VERIS) [20]. Both datasets included thousands of reported incidents spanning across various categories. Our model focuses on (1) Unauthorized access attempts, (2) Suspicious Denial of Service, and (3) Data Stealing Malicious Code, including ransomware instances. We filtered the incidents that conform to the categories and performed a qualitative assessment of the identified breach impacts. Lastly, for simplicity, we appllied an additional category that distinguishes the target component reported, service-based or client-based. We included only the service-based incidents. i.e., reported incidents that clearly targeted desktops and workstations were not included in defining tenant semantic structures.

We applied filters for training data accuracy. Filters for VERIS dataset included server workloads as indicated in Section 4.2.1, i.e., Authentication Server, Backup Server, Database Server, DHCP Server, Directory Server(LDAP, AD), Distributed control system, Domain Name Server, File Server, Mail Server, Mainframe Server, Web Application Server, and Virtual Machine Server[20]. Assets operating systems were filtered to Linux and Unix as such operating systems are more prevalent in servers than Windows, MacOSX, and mobile device operating systems.

VERIS dataset includes incident actions. We filtered the action types that fit the paper focus workloads. i.e. Brute Force, Cache Poisoning, Cryptanalysis, Fuzzing, and HTTP Request Smuggling attacks. We excluded Buffer overflow cases as such attacks can be prevented in deterministic methods and common in Windows-based operating systems[21]. The dataset size following the refinement is 5015 incidents. Table 1 summarizes the dataset we used for the training data.

Malware Category	modus operandi	Number of Incidents
Brute Force	Exhaustive effort of data encryption	946
Cache Poisoning	Corrupt data is inserted into the cache database e.g., DNS	894
Cryptanalysis	Exhaustive effort of data encryption	750
Fuzzing	Injects random bad data into an application to break it	946
HTTP Request Smuggling	Exhausting a proxy cache by sending HTTP requests	639
Data stealing malware	Data transmitting across unencrypted network	840

Table 1: Summary of datasets used

4.3.2 Removing the Tenant Semantics

Our approach attempts to detect anomalies in both control-plane and tenant activities that conform to suspicious patterns. In Section 4.3 Data Collection, we defined a categorical dataset that adheres to real incident data. This data applies to potential breaches for server-based workloads. We stipulate, for the purposes of this paper that such server-based workload will obey similar suspicious patterns when deployed in the cloud.

In this paragraph, we transformed the categorical dataset into a multivariate time series data that can be used for supervised anomaly detection. The multivariate set is comprised of general operating system observations that do not include any workload semantics but could be used for contextual anomaly detection. The contextual attributes are used to determine the position of an instance on the entire series. We showed that, based on collected incident data, the conversion of behavior patterns to multivariate time-series satisfies effective breach detection of any malware, conventional or polymorphic.

We gathered the operations reported in the incident reports (Table 1) and inferred about the operating system resources consumed during the malware lifespan. Table 2 depicts the relationship between the malware characteristics and operating system usage. Figure 4 describes a workload sample, video-on-demand. It shows the common pattern of the operating system resources usage that will be used as multivariate time series data sequences. The Evaluation section describes in more details the nature of the data and how it translates into meaningful time series data.

Malware Category	OS Resources Patterns
Brute Force	Extensive CPU, Memory, I/O to disk or network
Cache Poisoning	Extensive I/O to disk or network
Cryptanalysis	Extensive I/O to disk or network
Fuzzing	Network I/O Ingress
HTTP Request Smuggling	Network I/O Egress
Data stealing malware	Network I/O Egress

Table 2: Dataset Classification

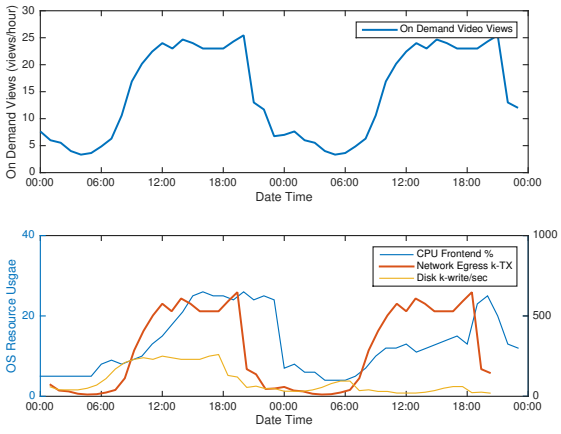

Figure 4: Workload semantics sample transformed into semantic-less training sequences

4.3.3 Prediction Methodology

We used the data gathered in Table 2 for formulating the anomaly detection problem of polymorphic malware[22]. The detection approach includes three distinct methods: (1) Detecting anomalous sequences in OS usages time series events, (2) Detecting anomalous subsequences within OS usages time series, and (3) Detecting anomalous OS usages events based on frequency. Let T denote a set of n training sequences based on OS usage generated by CSPs, SPs, and the Federation control plane. Also, S denote a set of m test sequences generated based on Table 2, we find the anomaly score $A(S_q)$ for each test sequence $S_q \in S$, with respect to T. T mostly includes normal OS usage sequences, while S includes anomalous sequences.

The semantic-less tool output produces a score for a scanned training sequence T using Regression. i.e., forecast the next observation in the time series, using the statistical model and the time series observed so far, and compare the forecasted observation with the actual observation to determine if an anomaly has occurred[22]. For simplicity, our model uses TensorFlow[23] for regression calculation. Our tool is not limited to that tool or the regression type.

5 Evaluation

We prototyped Cloud Federation system that mimics that properties analyzed in section 2, Cloud Federation. The prototyped system includes the component that is depicted in Figure 1. For the scope of the prototype, we enabled semantic-less metrics from both SPs and CSPs to improve correlation efficiency. CSP data sharing limits the effectiveness of any cyber analytical technique and, in practice will represent a compromise between improved cyber security and CSP privacy and confidentiality. With that proviso, in the following section, we evaluate a computer load coordination system component that manages on-demand streaming, generates T, a set of n training sequences based on OS usage generated by CSPs, SPs, and the Federation control plane. We chose on-demand video streaming as Video streaming is expected to constitute up to 85% of Internet consumers traffic within a few years[24]. Also, we showed that video-on-demand streaming follows a pattern of usage that can be monitored for breach detection that can help on-demand SP to seamlessly improve their consumer's privacy and provide their studio's safe e-commerce platform.

5.1 Experiment Planning

Below is a simulation of a cross-regional platform that is comprised of control-plane, workload-plane and coordinating components. This will be embodied in a resource allocation system (Kubernetes). This system provisions resources to be a priority of being near, users. The control-plane enables an effective compute resource provisioning system that spans across different public cloud providers and regions. Also, it collects operating systems usages for both the SP workload and control-planeThe coordinating components will accept user-workload demands as well as green energy availability from various regions and opportunistically seek to process streaming workloads using compute resources provisioned by green energy resources. The workload-plane will be comprised of edge streaming servers that process the end-user on-demand video streaming. It will be built on standard Apache HTTP[7] servers that run on the edge location.

The control-plane software infrastructure is based on Kubernetes[8], it facilitates internal discovery between Apache HTTP server instances so instances can connect across different cloud boundaries and regions. This architecture provides an open architecture that enables continuous monitoring. In a real world federation the data load may require several big data nodes and substantial compute capacity. This paper is a demonstration and proof of concept on a finite scale to permit model and parameter tracking and adjustment.

5.2 Execution

5.2.1 The System Preparation

The prototype experiment included the setup of three virtual datacenters deployed in different regions: (1) Central US, (2) West US and (3) East US. The clusters were sized based on US population distribution[9] by regions i.e. 20% for West US, 40% for East US and 40% Central US. The cluster sizes for West US, Central US, and East US are 3, 7 and 7 machines respectively. Each machine is standard 2-CPU cores with 7.5GB of memory.

The control-plane comprised of Kubernetes API server and controller-manager. The controller coordinator component will need to allocate resources across several geographic regions to different cloud providers. The API server will run a new federation namespace dedicated for the experiment in a manner that such resources are provisioned under a single system. Since the single system may expose external IPs, it needs to be protected by an appropriate level of asynchronous encryption[10].

For simplicity, we use a single cloud provider, Google Container Engine, as it provides a multi-zone production-grade compute orchestration system. The compute instances that process the user workloads are deployed as Docker containers that run Ubuntu 15 loaded with Apache HTTP server. For simplicity, we avoided content distribution by embedding the video content to be streamed in the Docker image. We ran 52 Docker containers that span across the three regions and acted as Content Delivery Network edges.

5.2.2 Baseline and Execution

The baseline execution included data populations for video streaming. The data population was achieved by the Kubernetes Jmeter batch jobs. The loader jobs goal is to generate traffic that obeys the observed empirical patterns depicted in

[7] Apache Web Server reference retrieved from https://httpd.apache.org
[8] Kubernetes reference retrieved from http://kubernetes.io
[9] US Population Distribution retrieved from https://www.census.gov/popclock/data tables.php
[10] Simulation code and data retrieved from https://github.com/yahavb/green-content-delivery-network

Figure 4. The system usage for both control-plane and SP capture, through cAdvisor, a kubernetes resource usage, and performance analysis agent. The agent, from every node in a cluster, populates system usage data to Heapster, a cluster-wide aggregator of monitoring and event data [11].

We labeled the system usage with the semantic-less dimensions, Network egress was measured by thousands of transmitted packets (k-TX), Disk writes per second (k-write/sec) and CPU usage per container (%). The Heapster aggregated the data based on the labels that are later pushed to centralized database, influxDB. We also used the influxDB HTTP API to inject randomized system usage data according to the three labels, CPU, network and disk usage. Those considered as the anomalous sequences $S_q \in S$. We used Figure 4 as a baseline sequence that randomized using NumPy[12]. The randomization followed the malicious usage patterns described in Table 2.

The execution required a TensorFlow session that looped through the dataset multiple times, update the model parameters and obtain the anomaly score $A(S_q)$ for each test sequence $S_q \in S$, on T. The breach and anomaly detection was performed using the following data streams and learning algorithms.

5.2.3 Limitations

We used influxDB because of its seamless integration with Kubernetes Monitoring system. However, our approach is not limited to influxDB or other database systems for that matter. We used TensorFlow for regression and anomaly score calculation. We did not use long training sequences. The maximum duration spanned across 48 hours. Training with longer sequences using long-running jobs and TensorFlow model checkpointing would improve our results. Our test content variety was limited and fixed. That might impact the generated tests sequences stability. Larger content variety would require longer training sequences for optimal detection.

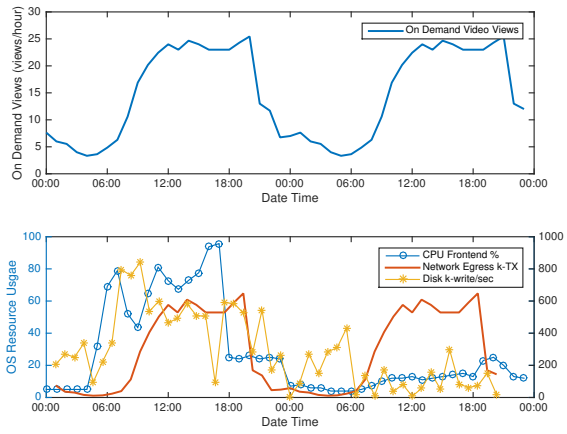

Figure 5: A Ransomware Anomalous Workload semantic-less sample

6 Analysis

The prototype included two core datasets, normal (Figure 4) and malicious (Figure 5). The CPU usage in the normal dataset fit the viewing patterns at the first half of the run. The second half required less CPU due to the caching mechanism applied in the Apache HTTP server that alleviates the need for the CPU when served through a cache. The Disk write pattern manifested similar content caching schema. The network egress ratio was not impacted by the caching schema.

The malicious dataset used the malware classification table (Table 2). Figure 5, shows a semantic-less behavior for ransomware malware that attempts to encrypt data while serving workload. Suspicious signals denoted by a star and o markers for CPU and disk write respectively. Based on the dataset classification, ransomware requires no network egress but CPU for data encryption and writing back to disk the encrypted payload. Our prototype included similar patterns depicted in Table 2 with a similar approach as done for ransomware.

Our model yielded a series of anomaly scores $A(S_q)$ for $S_q \in S$ anomalous patterns. We considered a potential breach of cases of anomaly and when regression produced a sufficient variability factor, i.e., a value that is not close to zero.

7 Conclusions

Security practices traditionally focus on prevention and tightening perimeter boundaries. However, with the advent of disparate, distributed, large scale, multi-tenant environments such as the proposed Cloud Federation, the traditional perimeter boundaries along with traditional security practices are changing. Defining and securing asset boundaries is more challenging and the system perimeter boundaries are more susceptible to breach. In this paper, we proposed a proactive approach for detecting a breach in a cloud workload. Such method requires no upfront investment from the monitored services. Upfront investments are often one of the main barriers to securing cloud service. Our tool eliminates such need and uses general system usage patterns that help to predict potential breach proactively.

Conflict of Interest The authors declare no conflict of interest.

References

[1] Leavitt, N. *Growth* **2009**, *27*, 15–20.

[2] Biran, Y.; Collins, G.; Dubow, J.; Azam, S. Federated Cloud Computing as System of Systems. 2017 Workshop on Computing, Networking and Communications (CNC) (CNC'17). 2017; pp 130–135.

[3] Biran, Y.; Collins, G.; Liberatore, J. Coordinating Green Clouds as Data-Intensive Computing. 2016 IEEE Green Technologies Conference (GreenTech). 2016; pp 130–135.

[4] Nelson, R. *EE-Evaluation Engineering* **2016**, *55*, 18–21.

[11] https://kubernetes.io/docs/user-guide/monitoring/

[12] Package for scientific computing with Python, numpy.org

[5] Kim, D.; Chung, H. R.; Thompson, P. R. Cloud-Based Automation of Resources. 2009; US Patent App. 12/634,050.

[6] Orr, R. J.; Abowd, G. D. The smart floor: a mechanism for natural user identification and tracking. CHI'00 extended abstracts on Human factors in computing systems. 2000; pp 275–276.

[7] Biran, Y.; Collins, G. Open compute-equipment design specification as a standard for cloud computing. Zooming Innovation in Consumer Electronics International Conference (ZINC), 2016. 2016; pp 70–75.

[8] Krutz, R. L.; Vines, R. D. *Cloud security: A comprehensive guide to secure cloud computing*; Wiley Publishing, 2010.

[9] Skorobogatov, S. P. Semi-invasive attacks: a new approach to hardware security analysis. Ph.D. thesis, Citeseer, 2005.

[10] others,, et al. Mayday: Distributed Filtering for Internet Services. USENIX Symposium on Internet Technologies and Systems. 2003; pp 20–30.

[11] Peng, T.; Leckie, C.; Ramamohanarao, K. *ACM Computing Surveys (CSUR)* **2007**, *39*, 3.

[12] Agrawal, T.; Henry, D.; Finkle, J. JPMorgan hack exposed data of 83 million, among biggest breaches in history. 2014.

[13] Krebs, B. *Krebs on Security* **2014**, *6*.

[14] Sidel, R. *Wall Street Journal, Sep* **2014**,

[15] Liu, Y.; Sarabi, A.; Zhang, J.; Naghizadeh, P.; Karir, M.; Bailey, M.; Liu, M. Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. USENIX Security. 2015; pp 1009–1024.

[16] Jaquith, A. *Security metrics*; Pearson Education, 2007.

[17] Chew, E.; Swanson, M.; Stine, K.; Bartol, N.; Brown, A.; Robinson, W. Performance measurement guide for information security. National Institute of Standards and Technology (NIST) Special Publication 800-55 Revision 1. 2008.

[18] Tufte, S. E. *Department of Management Science and Engineering, Stanford University, School of Information, UC Berkeley* **2015**,

[19] Center, C. C. *CERT Coordination Center* **2004**,

[20] Team, V. R. *Verizon Data Breach Investigations Report (DBIR)* **2015**,

[21] Wartell, R.; Mohan, V.; Hamlen, K. W.; Lin, Z. Binary stirring: Self-randomizing instruction addresses of legacy x86 binary code. Proceedings of the 2012 ACM conference on Computer and communications security. 2012; pp 157–168.

[22] Chandola, V. Anomaly detection for symbolic sequences and time series data. Ph.D. thesis, University of Minnesota, 2009.

[23] others,, et al. *arXiv preprint arXiv:1603.04467* **2016**,

[24] Biran, Y.; Collins, G.; Dubow, J.; Pasricha, S. Coordinating Green Clouds as Data-Intensive Computing. 2016: 3rd Smart Cloud Networks & Systems Conference (SCNS'16). 2016; pp 1–8.