Improving the Security of Cloud Based Systems

Joseph LeRoy
Graduate School of Arts and Sciences
Computer and Information Science Department
Fordham University
Bronx, New York 10458

***Abstract*—**

The cloud ecosystem has emerged as one of the most dominant technology services over the past decade. Just as much as smartphones have begun to dominate our personal lives, cloud service models have begun to replace on premises web servers, computers, databases, and file stores. With the rapid availability of shared resources on the cloud, many businesses are exposed to additional and more complex information security risks. Cloud service models such as Infrastructure as a Service, Platform as a Service, and Software as a Service introduce security risks that targets the entire scope of the OSI and TCP models, allowing a compromised host to potentially degrade the confidentiality, availability, and integrity of other systems on the network. Analyzing log files and presenting them in an easily digested manner is imperative to preventing data breaches and other security incidents. This research reviews several emergent technologies and presents an improved defense in depth approach, log analysis, and visualization to improve the security of cloud based systems.

I. Introduction

The expansion of open source software communities has accelerated greatly over the past decade with the emergence of cloud based systems, the growth and speed of networks, device storage and speed improvements, among other advancements in technology. A cloud based provider allows its customers to rapidly provision virtual server instances, with a generous share of server resources, and within any geographic location. This process is orchestrated using open source hypervisors such as Xen, used by Amazon Web Services, and Kernel-based Virtual Machine (KVM), used by Digital Ocean. There are also many closed source hypervisors such as zVM, developed by IBM, and ESXi, by VMWare. However, both of these hypervisors have an incredibly complex source code which allows for vulnerabilities and other malicious artifacts to degrade the security profile of cloud based systems as a whole.

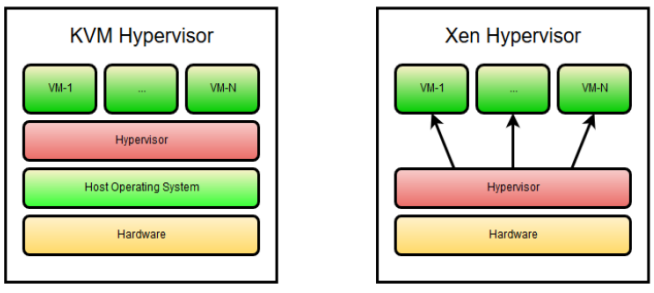

Fig. 1. KVM (type 2) and Xen (type 1) are two of the most widely used hypervisors for mnaging virtualized systems on cloud infrastuctures.

Another area of concern within cloud based systems is the integrity of the security profile due to the multi-tenancy, or sharing of resources and services, within each physical system. Each time a new virtual server is created an IP address, resources, and host name are assigned to it. There are two types of hypervisors that create and run virtual machines, listed in figures 1 and 2. The creation process begins at the physical hardware, moves through the host operating system and hypervisor (KVM) or solely the hypervisor (Xen), and then a virtual machine is created and logically separated from other hosts on the network. This new host coexists among many other hosts, some of which have no security controls in place. This presents a unique security concern due to the intrinsic value of the security profile being degraded by potential vulnerable software on other hosts that share the same resource pool. However, the cost savings provided through carpooling server resources has created an illusion of confidentiality and integrity, thus enabling malicious actors to exploit vulnerabilities on security negligent systems.

A. Cloud Computing Defined

Amazon Web Services has defined cloud computing as the on-demand delivery of IT resources and applications via the Internet with pay-as-you-go pricing. Cloud computing is basically a system for automating processes through self-service virtualized environments. This model enables individuals and businesses to rapidly deploy virtualized servers within a matter of seconds without incurring large costs when compared to traditional hosting. In many ways the difference is similar to renting an apartment versus purchasing a home. [1]

There are several characteristics that make cloud computing unique when compared to traditional datacenters. First, the systems that make up the cloud datacenter are required to be easily provisioned without aid from the hosting company. This process should be available during any hour of the day. Second, systems on the cloud hosting providers network should be available to all major cities and geographic locations. Next, the amount of storage, consumed or allotted bandwidth, CPU and RAM should be measured against an hourly pricing model to generate a variable monthly consumption bill. Finally, the hypervisor should allocate storage, CPU and RAM by resource pooling. A resource pool is a logical abstraction for flexible management of resources. Based on the resource pools consumption and the total amount of resources the hypervisor has to allocate, it divides the system into multiple virtualized systems and scales them accordingly. For example, if the hypervisor host has a 6 GHz clock frequency and 12 GB of RAM, you could have three resource pools with a 2 GHz clock frequency and 4 GB of RAM. [2]

Cloud computing is also comprised of two different models. There are service models, which include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). An IaaS provides the virtual machines, servers, storage, load balancers, and network components. A PaaS includes items such as databases and development tools, which are Service-Oriented Architectures, such as IBM Bluemix, Amazon Elastic Beanstalk, and Heroku. Theres also different deployment models, which define the type of access and controls each Cloud network contains. There are private, hybrid, community, and public Cloud deployment models. Community based Clouds are certified to host websites that are required to adhere to regulatory compliance laws. Examples are Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act (HIPAA), the Federal Information Security Management Act (FISMA), among others.

Nearly fifteen years ago Amazon Web Services introduced the concept of cloud computing and since then it has become the de facto standard for hosting digital content. Initially, the idea was to host virtualized systems for Amazons ecommerce website. It took them four more years to release the Simple Storage Solution (S3) and the pay-as-you-go model. Several months later, Amazon Elastic Cloud (EC2) was released, allowing businesses to use them as their hosting provider. In 2008, Google App Engine was released as a Platform as a Service, and a year later Microsoft launched Azure Beta. Cloud computing finally caught on as a mainstream component for businesses to rapidly deploy and scale infrastructures, platforms, and software depending on their budget and production requirements. [3]

II. Hacking Groups Target Cloud Based Systems

Some of the most popular countries where hacking groups and individuals originate are China, the United States, and Russia. China has roughly 721 million internet users compared to the United States which has 286 million users and Russia with 102 million. Coincidentally, these countries rank among the wealthiest and fastest growing economies in the world. The United States currently ranks as the wealthiest country with $48.73 trillion dollars worth of assets and China in second with $17.25 trillion (Wealthiest countries). However, Russia does not rank among the top three wealthiest countries, but it is however one of the fastest growing countries. Russias wealth grew by 253% from 2000 to 2015. The importance of this information that the economic status of a country might play a certain role in what fuels malicious actors, who they target, and what industries they seek to compromise. [4]

A. Intrusion Detections Defined

The SANS (SysAdmin, Audit, Network and Security) Institute defines an intrusion detection as the act of detecting actions that attempt to compromise the confidentiality, or availability of a resource (SANS Institute). There are three main categories to intrusion detection systems: network based, host based, and physical. This research primarily focuses on network and host based intrusion detection and analysis using applications and services such as Bro, Tripwire, Fail2Ban, and software to analyze the logs they generate. [5]

Network intrusions are an increasingly common occurrence which happen every second on cloud based systems. The affordability and ease of use has made cloud hosting an incredibly popular choice. However, this also allows both experienced system administrators who have a solid understanding of security and everyone else to share the same resource pool. This situation is comparable to having someone who just got their license to drive on the busiest highway in the world during rush hour traffic while in inclement weather. Statistically, there will be many accidents due to inexperience, just as there will be many vulnerable systems occupying resource pools with hundreds of other virtual machines on them. Naturally, this affects everyone using the service, just as a traffic accident causes disarray to overpopulated highways during rush hour traffic. This endangers the resiliency of cloud security in many ways and poses a challenge for businesses who provide cloud hosting and to those who use it as a service.

B. Hypervisor Vulnerabilities

Hypervisors have had their fair share of vulnerabilities in the past. The relevance of cloud based systems has made them an incredibly sought after target for ethical and malicious hackers. Due to the multitenancy nature of cloud based systems, arbitrary code can be executed to gain access to other virtual machines provisioned by the hypervisor. There are several types of vulnerabilities that have been discovered in recent years. One of the most talked about vulnerabilities was the VENOM (Virtualized Environment Neglected Operations Manipulation) security vulnerability or CVE-2015-3456. If a hacker were to exploit the VENOM vulnerability they would be able to escape the virtual machine and laterally move throughout the resource pool, and potentially move to other resource pools outside of its origin. [6]

#	CVE	Score	Vulnerability Types
1	CVE-2016-3961	2.1	DoS
2	CVE-2016-3960	7.2	DoS Overflow + Priv
3	CVE-2016-3159	1.7	+ Info
4	CVE-2016-3158	1.7	+ Info
5	CVE-2016-3157	7.2	DoS + Priv + Info

Fig. 2. Vulnerabilities found within the Xen hypervisor.

The Xen hypervisor technology employed by Amazon Elastic Cloud (EC2) has experienced about nine different vulnerabilities this year alone, as of May 20th. The most recent vulnerabilities affected the availability of services powered by the Xen hypervisor, as well as privilege sabotage and the ability to gain information through unauthorized channels.

In figure 3, an example of the VENOM (Virtualized Environment Neglected Operations Manipulation) security vulnerability shows how one vulnerable virtual environment, originating from Resource Pool A is able to escape to the hypervisor and then proceed to Resource Pool B and move to other virtual machines outside of its originating resource pool. This vulnerability is comparable to a quick spreading fire within an apartment complex without fire walls and partitions which prevent the expansion of damage to other apartments in the vicinity. [7]

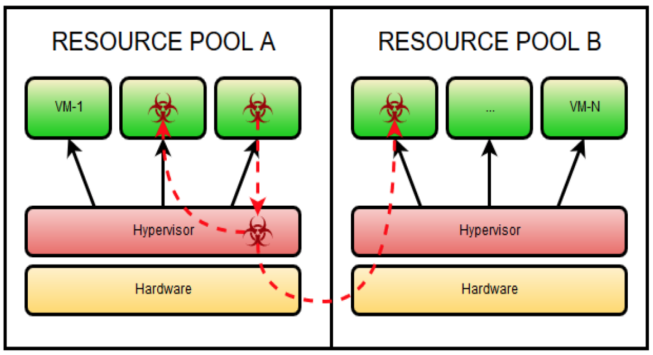

Fig. 3. VENOM hypervisor vulnerability proof of concept.

There are also other types of vulnerabilities common to cloud based systems that pose a huge security risk to both customers and providers of the service. Such being the case in side channel attacks. Hypervisor exploits that avoid normal segregation policies allow other virtual machines to collect artifact data left behind by virtual machines using shared hardware channels, such as the CPU cache. A recent exploitation coined CacheBleed exploits information leaks through cache-bank conflicts in Intel processors. A successful attack can recover 2048-bit and 4096-bit RSA secret keys from OpenSSL 1.0.2f running on Intel Sandy Bridge processors after observing only 16,000 secret-key operations . [8]

III. Proposed Countermeasures

There are many possible ways to improve the security of cloud based systems. In recent years the concept of Linux Containers (LXC) and software such as Docker Engine have come to light as a possible replacement for virtual machine environments, which currently exist on nearly a half-million Amazon Elastic Cloud (EC2) servers throughout the world. Eliminating the virtualization process allows the host operating system to reduce the overhead that comes with running a separate kernel and simulating all the hardware. Linux Containers employ many security enhancements over virtualized machines by removing the hardware layer and thus reducing the attack range for hackers to exploit. Several security features from the Linux Kernel are used in Linux Container technology, such as: Namespaces, AppArmor (Application Armor), Seccomp, Chroots, Kernel Capabilities, and Control Group functionality (cgroups). The preceding list of security features is non-extensive and will continue to grow as Linux Container technology matures. Kernel namespaces deal with the segmentation of the system, such as the groups, users, filesystem, and other access based controls. Control groups, or cgroups, deal with allocating processes, CPU utilization, memory utilization, disk input and output allowances, among others. [9]

There's also the concept of Unikernels, which is still a relatively new topic in the realm of microservices and cloud computing. In a nutshell, Unikernels work by using extracting libraries to build an extremely lightweight application. Through this process only certain aspects of the Linux kernel are utilizied and built into the application, thus reducing its attack surface, decomplexifying the code that has to be debugged, and thus making the application both faster and more secure. [10]

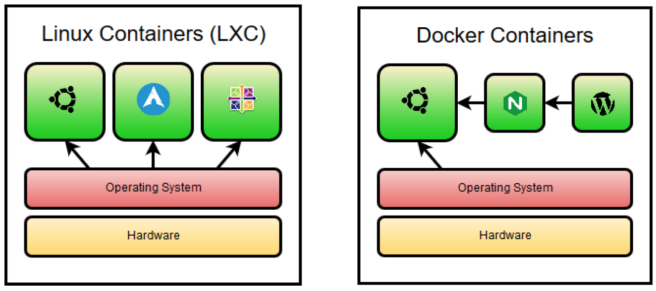

Fig. 4. LXC are vendor neutral, whereas Docker containers are read-only.

Another way to improve the security profile of cloud based systems is to integrate the Docker Engine into a currently existing environment. Docker containers, which run on top of Docker Engine, are lightweight by definition and allow developers to provide continuous integration and delivery. There are many ways that Docker can improve security on virtual machines, such as building a security centric container that contains a firewall, an intrusion detection system, and a log collection and visualization software stack. Docker can operate in both a stateful or stateless manner, but excels running stateless applications. For example, a container can be created to encapsulate an Nginx reverse proxy or an Elasticsearch powered database for storing log files.

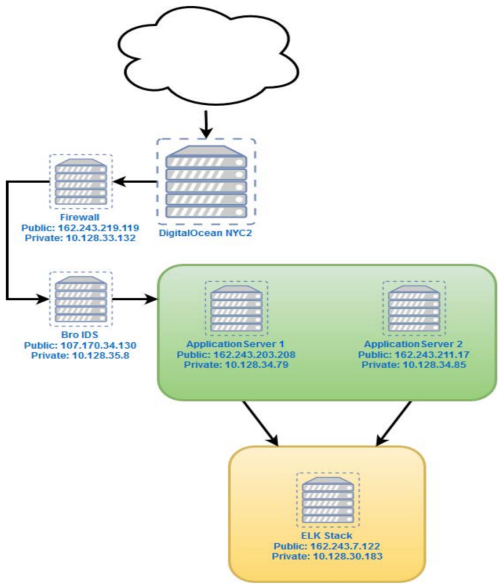

Fig. 5. Proposed network setup on Digital Ocean cloud IaaS.

An ideal environment presented in figure 5 could be implemented into an existing cloud based environment to protect application servers. Using some of the software and other technology listed in this paper, such as Bro and the ELK stack, the security profile can be greatly enhanced and thus protect not only the residing resource pool contents but potentially other data pools within the datacenter. Further research will explore an even greater approach to enhancing the

security on cloud based systems by using Docker to develop containers for each security mechanism, thus inheriting not only resource saving benefits but also Linux Container security enhancements.

A. Bro Intrusion Detection System

Bro is a very unique intrusion detection system that captures packets, inspects traffic, detects attacks, and records logs for each event it encounters. What Bro also does is detect intrusions, handles vulnerability management, file analysis, measure traffic, and deals with regulatory compliance monitoring for community cloud based systems. Bro logs can be sent into a data store and be analyzed by applications such as Splunk, the ELK (Elasticsearch, Logstash, and Kibana) stack, among others.

Using Bro as an intrusion detection system is an incredibly beneficial tool to understand network events and build or maintain situational awareness in respect to security alerts and anomalous activities. For example, when an instance of Bro is installed and configured on a host for network monitoring, a group of log files are dynamically created and populated with connection attempts, security notices, errors, and other security relevant information. After a set amount of time, the logs are compressed and stored in their own directory or data store for historical analysis.

Bro combines the principles behind many computer and network forensic tools, concepts, and best practices to form a powerful, open source solution for cloud based businesses to protect their assets while containing any vulnerabilities to a specific virtual environment, not others.

B. Tripwire Intrusion Detection System

Tripwire is a very powerful and easy to use host based intrusion detection system. A host based intrusion detection system does sits on each host to monitor inbound and outbound packets, file system changes, among others. Its the most powerful detection system for detecting insider abuse, such as disgruntled employees using administrative rights to destroy or modify files. During the configuration and installation process Tripwire creates an encrypted database where it stores a baseline image to reference against each file modification, addition, or deletion. The process of scanning an entire filesystem is relatively CPU intensive, so typically the service is run during non-peak hours. [11]

One of the unique security features that Tripwire offers is site key file encryption to ensure configuration files arent modified as well as local key encryption which prevents Tripwires binary files from being executed without permission.

C. Threat Intelligence

Threat intelligence is comprised of many different disciplines which in turn make it an entire field within the realm of information security. SANS describes CTI (Cyber Threat Intelligence) through a five stage process: victim, delivery, infrastructure, motivation, and actors. The FBI defines intelligence as any information that has been analyzed and refined so that it is useful to policymakers in making decisions specifically, decisions about potential threats to national security. [12]

The difference between information and intelligence is key to understand the importance of threat intelligence. Information is simply the log file files and events that occur on a network or within a system. Intelligence is the processed, analyzed, and qualified data from log files and events. By itself, Bro has many powerful threat detecting and vulnerability mitigating features. However, there are ways to make Bro and even more resilient security product. Critical Stack, a security company owned by Capital One, developed an intelligence marketplace to enhance the security event detection rate of Bro. The feeds are populated with malicious indicators, such as Tor exit nodes, malicious IP addresses, malware domain lists, C3 (command and control) systems, among others. There are over 1.3 million malicious indicators and over 100 intelligence feeds and theyre easily digested into Bros intrusion detection capability. [13]

Critical Stack

ID	NAME	LAST UPDATED	INDICATOR COUNT
94	snort.org-IP-Blacklist	04/30/16-03:17-am-(EDT)	37,411
25	ET-Known-Compromised-Hosts	04/29/16-07:15-am-(EDT)	1,020
24	Scam-Domains-(Fake/Malware/Drive-By)	03/03/16-10:59-am-(EDT)	5,181
23	Malware-Domains	04/30/16-03:15-am-(EDT)	20,847

Fig. 6. Critical Stack service output for Bro IDS.

Its important to note that threat intelligence always requires human intervention as well as a proper and sustainable business plan to ensure security is appropriately addressed, from the executive branch down. The tools and log analysis features that Bro provides allows us to gain intelligence on the malicious actors that target our servers. From there intelligence practices can be implemented to improve network resiliency. The action of examining log files, populating threat feeds, and interpreting events helps build reports such as the Verizon DBIR (Data Breach Investigation Report) and IBMs Cost of Data Breach Study.

D. Log Aggregation, Correlation, and Analysis

Log data is one of the most important resources security teams have at their disposal. Whether security teams are responding to an event or gathering intelligence, the process of aggregating the variety of logs is a challenging task, especially on cloud based systems. Normalizing log data so that it can be implemented into a data store is another important task. Each log, whether it be authentication logs, boot logs, MySQL logs, or web server logs, have very different ways of being interpreted. Each log speaks their own language and cannot be digested universally. Associating correlated data with values based on their function is essential to being able to create a mutual connection between the contents and some identifiable, perhaps known malicious value. It is essential to analyze data from a high-level perspective, such as a graph or a dashboard with multiple graphs, to better understand who, what, and why in a security incident.

UNIX based operating systems have become primarily focused around a select variety of Linux kernel distributions.

One of the most popular flavors of Linux is Ubuntu, which was released in October of 2004. In Ubuntu, logs are stored in the /var/log/ directory and typically contain a collection of authentication logs, boot logs, kernel logs, and if installed also contain web server logs. This helps system administrators troubleshoot issues that arise. It also provides the security community with breadcrumbs and artifacts to track hackers back to their origins and to understand how and what systems and applications were exploited during an incident.

1) Elasticsearch: Elasticsearch is a schema free, RESTful (Representational State Transfer) and JSON (JavaScript Object Notation) based distributed document store that requires close to zero configuration for a simple deployment. Its used by companies such as GitHub, Mozilla, SoundCloud, and Stack Overflow, just to name a few. The core of Elasticsearch is powered by the Apache Lucene library. Lucene allows Elasticsearch to gracefully process full text search and indexing of large amounts of data. [14]

One of the many powerful features of Elasticsearch is that it is easy to setup a node cluster, which ensures reliability and the overall health of the data store, which contain important log files and other security information. In a production level environment at a SOC (Security Operations Center) it would be imperative to ensure data is always available and in near real-time. You can accomplish this in multiple ways. The first way is through data duplication and the other is through data partitioning. One of the Amazon AWS Elastic Cloud (EC2) clusters that I configured using the ELK stack uses a data duplication method. Through this process the same exact log data is split between two nodes, communicated between each other to ensure they both have the same data, and the load is split between each other using the round robin scheduling algorithm by implementing an ELB (Elastic Load Balancer). This is just a traffic load balancer for incoming web requests. Another way to distribute data using Elasticsearch is by using a data partitioning method. This simply allows the node requesting data to search over a wide number of data stores rather than just one. A data partitioning setup is typically implemented when theres a huge amount of data being stored, which surpasses the requirements of any one system.

Elasticsearch is very easy to manipulate and the core of its power is accomplished through a plugin based ecosystem. Since Java is required, custom JAR files can be developed as plugins to increase the functionality of Elasticsearch. Site plugins can also be developed to increase functionality and require some knowledge of HTML, CSS, and JavaScript. Additionally, a mix between Java and site plugins can also be developed. The only difference between the two main categories is that Java plugins are required by all nodes running Elasticsearch, whereas site plugins can exist independently.

2) Logstash: Logstash is an application that follows a three stage process: receive inputs from server logs, use filters to normalize data, and output it so that Elasticsearch can quickly store it as a JSON file. Its important to note that Logstash can act as its own server instance rather than residing on the same instance as the data store, as can all parts of the ELK stack. This setup would be best in a production environment, whereas having the entire ELK stack on one server instance would be beneficial to learn the internal components of each application or for demonstrating a proof of concept. [15]

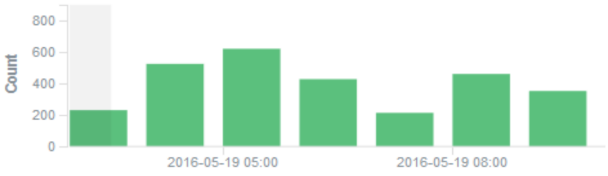

Fig. 7. SSH authentication failures by hour for two virtual machine hosts on Digital Ocean.

In the case of a production environment, Logstash would need to have a way to communicate log files back to Logstash for further processing, storage, and analysis. To do this we use system daemons called Elastic Beats. Elastic Beats are lightweight daemons written in the Go programming language. Elastic Beats reside on each endpoint that requires monitoring. They communicate using TLS 1.2 encryption to ensure the confidentiality of the log data remains intact while in transit.

3) Kibana: Kibana is a frontend application that displays a visual representation of post processed system log contents. It works by connecting to Elasticsearch and encrypting log data sent from the application servers using a public and private key pair. One of the powerful features to Kibana is its ability to tie into various departmental functions of a business. Kibana allows you to display log data in the following formats: area charts, data tables, line charts, markdown widgets, metrics, pie charts, tile maps, and vertical bar charts. One of the most interesting visualization features is displaying geographic IP addresses as latitude and longitudes. By translating the IP address to a geographic location, Kibana can display the geographic location of the host generating the log entry, associate it to an event. In return, this increases the situational awareness through deductive reasoning. If thousands of attacks originate from Russia, firewall rules to block or filter the traffic should be implemented and enabled. [16]

Through extensive modification, an internal network can be geographically mapped by region, individual location, or department, thus allowing for a more robust endpoint protection system. This would greatly aid in an incident response situation to determine where a vulnerable system exists on the network, and from which part of the office it resides. The heart of Kibana is written using mostly JavaScript, and is completely open source on GitHub. It can be modified to fit the needs of any business and has many features which make it a great application to use for a SIEM (Security Information and Event Management) system. In a typical setup, Nginx will be used as a reverse proxy to allow external web access to Kibana, which connects to the entire ELK stack. This allows for more stringent security controls, such as basic authentication of HTTP users and enforcing SSL encryption. This should be sufficient for a nonproduction environment.

4) Elastic Beats: Elastic Beats are lightweight daemons written in the Go programming language. Theyre used to collect and ship logs from various endpoints to Logstash. File specific Elastic Beats such as File Beat search through log paths locations specified in a configuration file, matches the log with an input type and a document type so that the logs can be easily interpreted by a filter. [17]

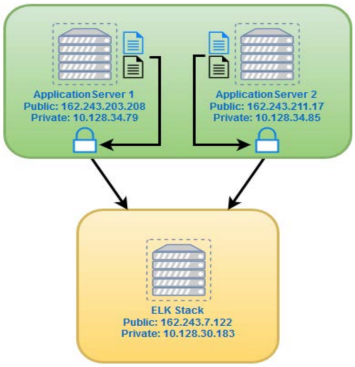

Fig. 8. Log file encryption and secure transport using TLS to send log data from each application server to the central ELK server.

The log data is sent to the ELK stack server, starting with Logstash. Its important to generate a proper encryption channel.

IV. Conclusion

This research has explored the incredibly vast ecosystem of cloud based computing and how it has imposed greater security risks to businesses and individuals. By using threat intelligence feeds to enhance the function of intrusion detection systems, we can employ a certain due diligence based approach to prevent security incidents. Using some or all of the features and log outputs from firewalls and intrusion detection systems, as well as native operating system logs, we can store inbound and outbound packets, flow data (consolidated packet captures), and other forensically important artifacts in the ELK (Elasticsearch, Logstash, and Kibana) stack for visualization.

There's a lot of research and effort being put into the containzerization movement, through native Linux Containers found within modern Linux distributions, Docker, and Unikernels. By implementing a segregated way to isolate the system's resources and access controls, while preventing common hypervisor vulnerabilities, reduce the amount of resources required has many inherited security benefits. Many of which are outlined in the preceeding sections of this paper, but also many additional whitepapers and research can be found on popular opensource communities such as Github, which are contstnatly updated and fully transparent. One useful set of guidelines, practices, and research can also be found in NCC Group's whitepapers, specifically toward improving cloud security by using Linux Containers. [17]

While security and vulnerabilities will always exist on network based systems, using emergent technology such as Linux Containers and Docker Engine to power the aforementioned software we can generate momentum toward making data breaches, software and hardware exploitation, phishing attacks, and other malicious actions less impactful to the cloud community and the internet as a whole.

References

[1] "What is Cloud Computing? - Amazon Web Services", Amazon Web Services, Inc., 2016. [Online]. Available: http://aws.amazon.com/what-is-cloud-computing. [Accessed: 30- Jun- 2016].

[2] "VMware vSphere 4 - ESX and vCenter Server", Pubs.vmware.com, 2016. [Online]. Available: https://pubs.vmware.com/vsphere-4-esx-vcenter/index.jsp. [Accessed: 30- Jun- 2016].

[3] "Amazon Media Room: Press Releases", Phx.corporate-ir.net, 2016. [Online]. Available: http://phx.corporate-ir.net/phoenix.zhtml?c=176060p=irol-newsArticleID=503034. [Accessed: 30- Jun- 2016].

[4] "Top 10 Countries with Most Hackers in the World", Country-Detail, 2015. [Online]. Available: http://www.countrydetail.com/top-10-countries-with-most-hackers-cyber-criminals/. [Accessed: 30- Jun- 2016].

[5] "SANS - Information Security Resources", Sans.org, 2016. [Online]. Available: https://www.sans.org/security-resources/idfaq/. [Accessed: 30- Jun- 2016].

[6] "CVE -CVE-2015-3456", Cve.mitre.org, 2016. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2015-3456. [Accessed: 30- Jun- 2016].

[7] "VENOM Vulnerability", Venom.crowdstrike.com, 2016. [Online]. Available: http://venom.crowdstrike.com/. [Accessed: 30- Jun- 2016].

[8] [8]"CacheBleed: A Timing Attack on OpenSSL Constant Time RSA", Ssrg.nicta.com.au, 2016. [Online]. Available: https://ssrg.nicta.com.au/projects/TS/cachebleed/. [Accessed: 30- Jun- 2016].

[9] "Linux Containers", Linuxcontainers.org, 2016. [Online]. Available: https://linuxcontainers.org/. [Accessed: 30- Jun- 2016].

[10] "Unikernels - Rethinking Cloud Infrastructure", Unikernel.org, 2016. [Online]. Available: http://unikernel.org/. [Accessed: 30- Jun- 2016].

[11] "How To Use Tripwire to Detect Server Intrusions on an Ubuntu VPS — DigitalOcean", Digitalocean.com, 2016. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-use-tripwire-to-detect-server-intrusions-on-an-ubuntu-vps. [Accessed: 30- Jun- 2016].

[12] "Who's Using Cyber Threat Intelligence and How?", Sans.org, 2016. [Online]. Available: https://www.sans.org/reading-room/whitepapers/analyst/cyberthreat-intelligence-how-35767. [Accessed: 30- Jun- 2016].

[13] "Critical Stack Intel", Intel.criticalstack.com, 2016. [Online]. Available: https://intel.criticalstack.com/. [Accessed: 30- Jun- 2016].

[14] "elastic/elasticsearch", GitHub, 2016. [Online]. Available: https://github.com/elastic/elasticsearch. [Accessed: 30- Jun- 2016].

[15] "elastic/logstash", GitHub, 2016. [Online]. Available: https://github.com/elastic/logstash. [Accessed: 30- Jun- 2016].

[16] "elastic/kibana", GitHub, 2016. [Online]. Available: https://github.com/elastic/kibana. [Accessed: 30- Jun- 2016].

[17] "elastic/beats", GitHub, 2016. [Online]. Available: https://github.com/elastic/beats. [Accessed: 30- Jun- 2016].

[18] "Understanding and Hardening Linux Containers", Nccgroup.trust, 2016. [Online]. Available: https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2016/april/understanding-and-hardening-linux-containers/. [Accessed: 30- Jun- 2016].