Cloud Security: Attacks and Current Defenses

Gehana Booth, Andrew Soknacki, and Anil Somayaji

Abstract—**This paper presents a high-level classification of current research in cloud computing security. Unlike past work, this classification is organized around attack strategies and corresponding defenses. Specifically, we outline several threat models for cloud computing systems, discuss specific attack mechanisms, and classify proposed defenses by how they address these models and counter these mechanisms. This examination highlights that, while there has been considerable research to date, there are still major threats to cloud computing systems, such as potential infrastructure compromise, that need to be better addressed.**

Index Terms—**Cloud Computing, Security, Virtual Machines**

I. Introduction

CLOUD COMPUTING is now the foundation of most Internet usage. Email, search engines, social networks, streaming media, and other services are now hosted in "the cloud"—large collections of commodity servers running coordinating software that makes individual hosts largely disposable. While cloud computing has lowered costs and increased convenience, the accessibility and centralization of cloud computing also creates new opportunities for security breaches.

Many security researchers have studied various aspects of cloud computing security from both an offensive and defensive perspective. In this paper we give a high-level classification of this work in order to examine to what degree proposed defenses can address different kinds of cloud-specific attacks. Specifically, we organize the cloud security literature into five areas: colocation denial of service, colocation breaches of confidentiality, data integrity and availability, data confidentiality, and infrastructure compromise. As we will show, while there has been significant progress, there remain major shortcomings in cloud defenses, even from the perspective of published research. While there have been other cloud security surveys and classifications published, ours is the first one organized around cloud-specific attacks and defenses. Our hope is that this survey can help guide researchers to work on areas of cloud security that have been less studied.

The rest of this paper proceeds as follows. Section II gives a bit of background on cloud computing. Section III describes our assumptions about the nature of threats against cloud computing as opposed to other computing platforms. Then in Sections IV, V, VI, VII, and VIII, we survey published attacks and defenses regarding our five attack areas–colocation denial of service, colocation breach of confidentiality, data availability and integrity, data confidentiality, and infrastructure compromises. Section IX details related work to our survey. Section X discusses the limitations we found in the literature and potential areas for future research.

II. Cloud Computing

As outlined by Mel and Grance [1], cloud computing generally has five characteristics:

1. **Resource pooling**

   The provider's resources are pooled and shared between multiple customers.

2. **Broad network access**

   These resources are accessible through standard network protocols over the Internet.

3. **Rapid Elasticity**

   In a matter of minutes resources may be provisioned to scale out and released to scale in.

4. **Measured service**

   The provider measures and generally charges for usage of CPU, memory, disk, network bandwidth, or other resources.

5. **On-demand self-service**

   Resources can be provisioned via automated mechanisms

While every type of cloud service has these characteristics at its core, the various service models differ drastically in both form and function. We focus on three main service models: infrastructure as a service, software as a service, and platform as a service. Infrastructure as a service (IaaS) is the most basic service model for delivering cloud capabilities. Typically the consumer is given access to processing, storage, networks, and other resources necessary to run and/or deploy arbitrary software in a form that is close to having on-demand access to an arbitrary number of network-connected servers. An arbitrary number of "virtual servers" are multiplexed onto the providers' fixed number of physical hosts, generally using virtual machines (VMs) running on hypervisors. An example of IaaS is Amazon's Elastic Compute Cloud (EC2) service:

This work was supported by Canada's Natural Sciences and Engineering Research Council (NSERC) through the ISSNet Strategic Network and Discovery Grants programs.

The authors are with the School of Computer Science, Carleton University, in Ottawa, Canada. (contact email: soma@scs.carleton.ca)

.

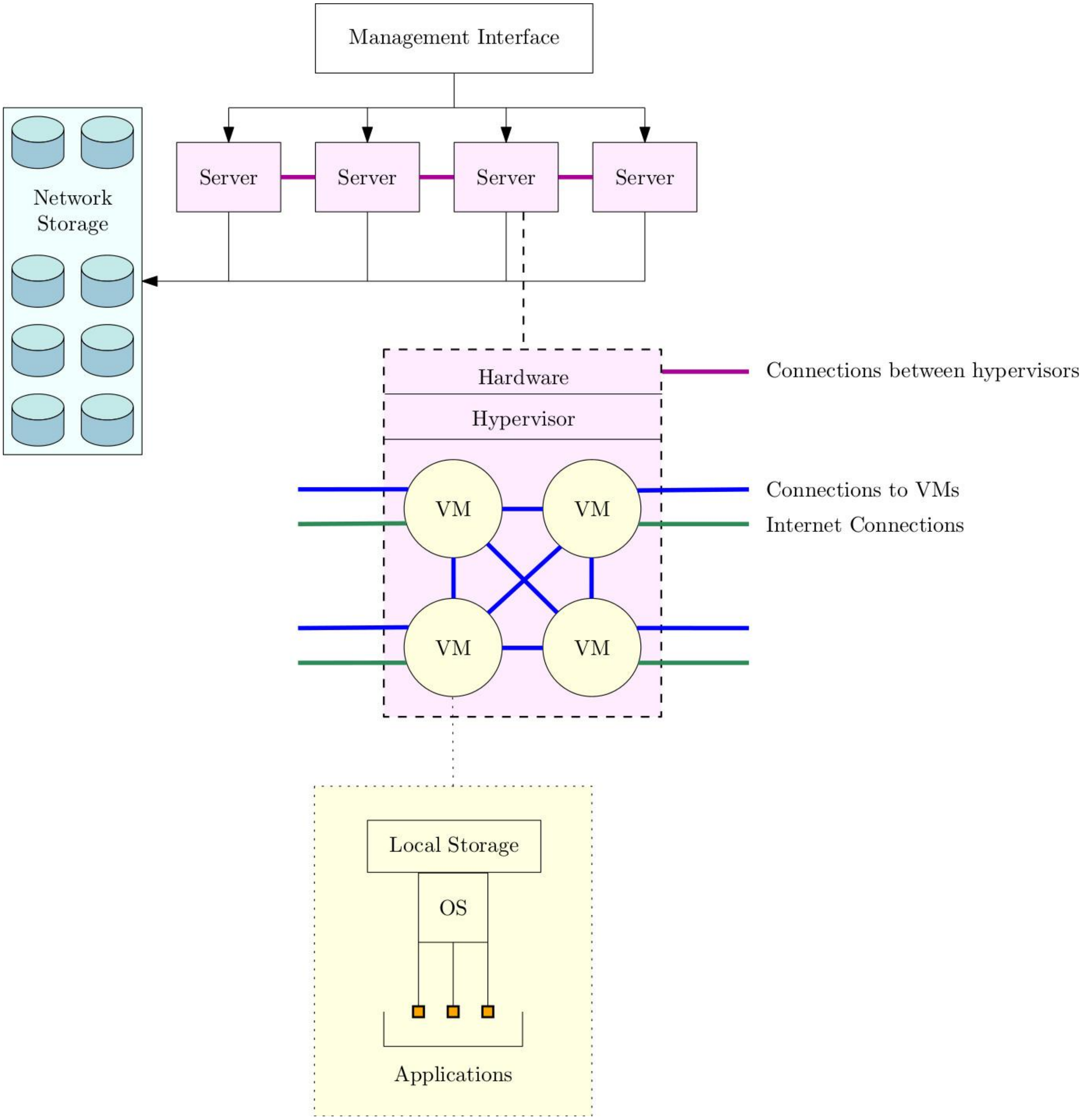

Fig. 1: The standard architecture of cloud computing infrastructure. Note this same infrastructure can be used to provide Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

the consumer is given access to an EC2 "instance" (a VM) for a period of time to be used as a resource for whatever purpose the consumer wishes. Another example of IaaS would be Amazon's S3 service: the consumer is given access to low-latency data storage that is accessible from any location via the Internet.

With Platform as a Service (PaaS), the consumer has access to computational platforms including operating systems, programming language execution environments, databases, web servers, etc. These combined services are mainly used by developers who use the provided platform to run and test their software solutions on a cloud infrastructure without the overhead of maintaining the underlying software or hardware. Google App Engine [2] is an example of PaaS which is utilized for developing and hosting web applications within Google-managed data centers. The developed applications are sandboxed and run across multiple servers for testing. Amazon Web Services (AWS) Elastic Beanstalk [3] is another PaaS system where clients are able to deploy their created or acquired applications on a virtual machine (e.g., in the form of a runnable jar) in order to test and deploy it. AWS Elastic Beanstalk is built on top of Amazon EC2, S3, and other parts of Amazon's IaaS offerings.

In the software as a service (SaaS) service model, the provider installs and operates application software on a cloud infrastructure. Clients may then access the software using a service-specific client software or a generic web browser interface. As with PaaS, SaaS providers are often consumers of IaaS. An example of this would be Dropbox. Dropbox allows clients to store their data and access it from any location via either the Dropbox website or the software one can install on their personal machine. Note that Dropbox has its software running overtop Amazon's S3 service for mass data storage [4]. Netflix is also a company that both provides and consumes cloud computing services. Netflix allows consumers to access movies and TV shows from any location via their website or installed application. While providing this service, Netflix layers their software and functionality atop Amazon Web Services [5]. Another example of SaaS is the Google search engine. Clients access their search engine through a standard web interface and are able to search the Internet for answers, solutions, etc. However, contrary to the traditional approach to cloud computing used by companies such as Dropbox and Netflix, the Google search engine uses its own infrastructure and does not employ VMs [6]. In other words, the Google search engine does not layer its software atop a cloud infrastructure that is already in place, such as Amazon EC2. Facebook also falls into the SaaS model, but follows the same vein as the Google search engine in that Facebook has defined, implemented, and uses its own infrastructure without utilizing VMs or third-party cloud services such as Amazon EC2 [7].

Thus while these different service models have different economic, administrative, and consumer experiences, they all share a common architecture that is typically something akin to that shown in Fig. 1. Variations include the use of local versus remote storage pools and the degree to which hypervisors are employed to host full operating systems or the use of other mechanisms, such as language-based virtual machines, to separate customers. For example, as we mentioned, Google and Facebook do not use virtual machines because users of their services—even services such as Google App Engine [8]—are only allowed to run certain types of software running in restricted environments. Most others in cloud computing, however, either provide or make use of collections of virtual machines connected to storage pools.

It should be noted that the instances of these models which we have detailed above are all examples of public clouds—clouds that are made for consumption and utilization by the general public—rather than private clouds which have been internally constructed for use by one company or organization. As we will see, this key difference enables several new kinds of attacks. Throughout the paper we utilize Amazon and its cloud services as the main example of a public cloud provider and platform. We do this as Amazon is the most popular platform at present for a public cloud. However, they have many prominent competitors including RackSpace, CloudSpace, and Microsoft Azure. While their technology stacks differ, significant overlap in functionality and increasing interface standardization means that customers can migrate between services and develop systems that span providers. Thus the attacks and defenses outlined in the paper, while taking Amazon as their main example, also apply to other public cloud providers. These attacks and defenses are also applicable to private clouds, but only to the extent that attackers can gain access to the cloud infrastructure.

III. Security and the Cloud

Cloud computing infrastructure is, in principle, subject to all of the threats that standard server computing infrastructure is. Web servers can be compromised with cross-site scripting vulnerabilities; databases are subject to SQL injection attacks; operating system kernels can be compromised by machine code injection. Here, however, we are concerned with ways in which cloud-based systems are different from traditional servers from a security perspective.

In the following sections, our focus is on attacks that only make sense in a cloud computing context, as these are the new risks that arise when transitioning to the cloud. We should note, however, that cloud-based systems potentially do have some security advantages. Cloud providers can automate and provide as a service many standard systems administration tasks such as backups, software patching, and network monitoring. Virtual machines may be "reinstalled" very quickly through automated provisioning, allowing virtual machines that have been compromised to be more easily replaced than servers running on raw hardware. The security state of virtual machines and their associated storage may also be monitored externally (outside the scope of the guest VM's potentially untrustworthy applications and operating system) for malware by scanning files and even having the hypervisor directly detect intrusions in running VMs using introspection techniques [9]. While these are potentially useful, they are also things that could be implemented outside the cloud. The attacks and defenses we discuss in the rest of this paper, however, are all unique to applications running in the cloud, particularly public clouds.

In the rest of this paper we make the following assumptions. We assume that cloud applications are run within virtual machines running on hypervisors with local storage and access to remote network storage as shown in Fig. 1. The target (victim) is assumed to have one or more VMs in the cloud. We assume that the attacker is either on the public Internet connecting to the targeted VM or that the attacker has a VM with the same cloud service as the targeted VM. For some attacks, we further assume that the attacker has a VM on the same host—running on the same hypervisor—as the target. In all of our scenarios, we assume that the target's software—their applications and operating system—are otherwise secure. Thus the attacker is primarily taking advantage of the fact that the target is making use of a cloud computing infrastructure.

IV. Colocation: Denial of Service

In a cloud infrastructure, CPU, RAM, disk, and network bandwidth resources are shared between users. As such, if an attacker consumes a large amount of resources, all other customers that share the same physical resources will notice a decrease in performance. If severe enough, this decrease constitutes a denial-of-service attack. Customer applications may be migrated to other part of the cloud infrastructure with less resource contention; however, even the largest of providers have finite resources.

Cloud providers employ a variety of strategies to partition resources in such a way that such denials of service—whether accidental or deliberate—are less likely. Providers such as Amazon divide their cloud into "availability zones" that are designed to fail independently. To maximize uptime, developers must replicate their applications in multiple zones and allow fail-over between them. Within data centers, networks are partitioned by routers and network-level quality-of-service mechanisms [10]. Hypervisors such as Xen implement "fair share" CPU schedulers that give at most a fixed portion of a node's CPU and I/O bandwidth [11].

These partitioning strategies are not perfect, however, and it is possible to cheat, allowing users to exceed their allocation. For example, hypervisor schedulers can be manipulated into misallocating CPU resources [12]. Even if a customer's virtual machine gets its allocated share of resources, it may not get them in a timely fashion, causing increases in network response latency. Such increases in latency can be particularly harmful for cloud-hosted web applications. Thus one area of research to further explore is in improving latency under load [13].

Another key defense strategy is economic: they charge for resources used. Providers use existing metrics such as peak network bandwidth and storage consumption to measure and charge customers. Where metrics were not so readily available, such as CPU resources, providers have created new ones: Amazon's EC2 compute unit (ECU), for example, is defined as the power of a 1.0−1.2 GHz 2007-era AMD Opteron or Intel Xeon Processor [14], [15]. While the consumer is utilizing cloud services, this metric is monitored, most typically by a VM Monitor. Once the consumer has used the resources that their SLA has provided for (i.e. once the consumer has expended the amount of cloud services that they have initially agreed to and paid for), they are seen to be in violation of their SLA with the cloud provider. The cloud provider then utilizes a gradual formula to determine how much the SLA has been violated, meaning that the more the SLA is violated over time, the larger the penalty for the consumer will be. This penalty is typically in the form of financial recompense. With this kind of metering in place, resource-based denial of service then often becomes a matter of fraud, either of evading the metering mechanisms or paying for services using stolen credentials (e.g., credit card numbers).

As providers get larger and better able to manage their resources using partitioning and economic mechanisms, pure colocation denial of service is becoming increasingly infeasible. Other kinds of attacks, however, are still very possible.

V. Colocation: Breach of Confidentiality

With colocation-based breaches of confidentiality, attackers attempt to use colocation in order to compromise the confidentiality of a VM. Information about the data stored inside a VM can be inferred by noticing patterns of resource usage, particularly CPU usage. Such resource usage can be inferred through resource contention with a co-located attacker virtual machine.

For example, Ristenpart, et al. [16] outlined a series of attacks against the Amazon EC2 service. They would start up several instances (usually over 100 to gain the desired results) with the aim to hit a target. The first of these types of attacks is the gambler attack. This attack attempts to hit a target, any target, and compromise it. The second of these attacks is the sniper attack. During a sniper attack, attackers compromise a single, specific target. Once the attacker has chosen his intended victim(s), the attacker then attempts to influence the victim to react in a way that they can predict so that the attacker may extract information. Both of these attacks take advantage of the fact that many VMs will run on the same node (host). The creation and use of several hundred instances is meant to make it feasible for the attacker to land in the targeted (either arbitrarily or specifically) VM.

Attackers can use multiple ways to determine if they have landed on the target's node. One is a network-based strategy where simple IP scans are used to determine if the attacking instance and targeted instance share the same administrative IP address (e.g., the IP address of their Xen Dom0 instance). Another is to check whether there is a low latency network path with the target (i.e., whether packets can be exchanged with minimal transmission delay). Or, the attacker can check to see whether accesses to the target increase the rate of cache misses in the attacker's VM; if it does, then they are sharing hardware [16]. Once the attacker shares a node with the target, timing and cache interference effects between VMs can be further used to extract information from the target, such as OS information [17] and even cryptographic keys [16], [18].

There are a number of stages where defenses can help prevent these kinds of attacks. One is to prevent the attacker from sharing hardware. While this sort of protection can be gained by moving to a private cloud, even public cloud providers can give some of this type of protection by guaranteeing exclusive access to nodes (for an extra fee, of course). The provider can also randomize their VM distribution schemes to reduce the probability of attacker/target co-location. But if we assume the attacker will be running on the same node (as is likely for a gambler attack), then we must minimize potential communication channels between VMs.

The next step would be for cloud providers to block the side channels that attackers may exploit. There are three approaches that a cloud provider could take for this with regards to the cache-level attacks. The first of these is to guarantee exclusive access to CPU caches (L1, L2, or L3). If a consumer has exclusive access to their caches, they can potentially detect hostile intrusions to their VMs via analysis of cache level noise [19]. If the consumer is barred exclusive access to the entire cache, they can be granted exclusive access to a portion of the cache through cache partitioning, for example through cache coloring [20]. In partitioning the cache, the amount of cache information overflow that may cause information leakage is greatly reduced as the attacker may no longer monitor what is being ejected from or altered within the cache by other VMs. Finally, if the consumer is not guaranteed exclusive access to the cache in any way, they may then be able to monitor the cache to determine if an attacker is attempting to extract any information. One cache-level attack strategy involves priming the cache with a large amount of temporary files. Such patterns of of malicious behavior can be detected through VM introspection [21]. Also, cryptographic timing attacks can also be mitigated by reducing the precision of the system clock, as these attacks require very precise timing [22].

VI. Data Availability and Integrity

Here we consider the following problem: how can a customer trust that a provider has the data they are supposed to be storing? Specifically, how does a customer know whether their data is accessible and has not been corrupted? Currently, cloud providers only guarantee uptime in their service level agreements; they do not explicitly guarantee data integrity or availability [23]. As such, cloud providers are under no obligation to prevent or notify the consumer of data corruption or loss of data availability.

The customer can, of course, verify data accessibility and integrity by manually accessing all remotely stored information. Computational and bandwidth constraints, however, make conventional implementations of such operations prohibitive in most contexts. Research into this area focuses on ways to make customer checks of data more feasible.

For example, the protection mechanism of the High Availability and Integrity Layer (HAIL) [24] attempts to do this by implementing similar functionality to Redundant Array of Independent Disks (RAID), in that data is mapped onto multiple (virtual) disk drives using a combination of two cryptographic functions–Proof-of-Retrievability [25] (POR) and Proof of Data Possession [26] (PDP). POR is a cryptographic function meant to enable a prover (the cloud provider) to demonstrate to a verifier (the consumer of cloud services) that a certain file is retrievable. This is done with the use of a small checksum, giving a high efficiency benefit as only a very small amount of data, not an entire file, needs to be transmitted. PDP is meant to show that a file stored in the cloud has not been altered or modified and that the consumer has access to said file without the need to fully download it.

While the POR and PDP functions of HAIL can greatly reduce the bandwidth required to verify data availability and integrity, they both have high enough computational complexity that they are not feasible to be implemented on today's systems. Fortunately mechanisms for integrity and availability monitoring of cloud providers is an active area of research [24], [25], [27]. Practical, deployed solutions to customer checking of data, however, still remain to be developed.

VII. Data Confidentiality

While consumers of cloud services desire cloud providers to both store and serve their data, they do not necessarily want the cloud providers to have free access to their data as this would be a breach in confidentiality. Yet today there are currently no default methods in place to prevent cloud providers from having free and ready access to the data they are storing and serving. Malicious providers of cloud services could freely peruse the data they are given access to by clients as, by default, all data is stored in the clear.

The natural solution is to encrypt cloud resident data. Simple encryption, however, is not so straightforward to implement. One reason is that cloud-based virtual machines are generally used to process cloud-resident data. They must have access to the data, so even if the data is encrypted the cloud provider also, implicitly, has the key to the data (in the form of the running virtual machines). Thus, encrypted storage for completely cloud-based systems has inherently limited benefits.

Another problem with encrypted storage in the cloud is that naive implementations of file-level or block-level encryption are all subject to traffic analysis. In other words, data access patterns themselves can breach confidentiality even if all of the data being accessed is encrypted. To prevent traffic analysis, data from different files have to be mixed together in terms of both reads and writes in such a way as to confound traffic analysis without incurring too much overhead.

Another aspect of this traffic analysis problem is that file metadata, in addition to file data, must be encrypted to prevent monitoring of specific users or groups. This problem is particularly significant for enterprises wanting to give selective access to their cloud-based data. Multiple privacy-preserving access control mechanisms have been proposed. An example would be the system proposed by Raykova et al. [28]—a double layered Access Control List (ACL) would be in place whereby one layer would specify what files a cloud provider should and should not have access to and a secondary layer for the user, when their VM is up and running, to provide and specify fine grain access. The general idea behind this being that, just because user data is being stored on the cloud does not mean that the cloud provider should have access to said data, nor should they have knowledge of who is able to access the data at all. Furthermore, as the user data is now stored on a public cloud, not a private VM or network, the ACL mechanisms need to take into account the large number of users accessing the cloud services and, as such, should provide much finer grained controls. Yu et al. [29] and Wang et al. [30] have also outlined ACL systems that could be applied to the cloud. These proposals do not appear to currently be mature enough, however, to be implemented on current systems. We see there is a big opportunity in this area for solutions that provide sufficient confidentiality while being practical for current providers, applications, and customers to use.

VIII. Infrastructure Compromise

Infrastructure compromise is the most unexplored threat area in cloud security. However, it is also the attack with the highest amount of payoff: If successful, the attacker gains a level of privilege akin to attaining root access on a machine. The weak point exploited by the attacker is a management interface of the provider (internal or external) rather than a direct attack against the cloud infrastructure. Even though it is the provider's resources that are exploited, the attack affects consumers as well.

Attacks against the management interface of cloud services are mostly unexplored. Amazon's cloud service provides a Simple Object Access Protocol (SOAP) and REST-based interface, with the SOAP interface being defined by an XML schema [31]. One aspect of the attack outlined by Somorovsky et al. [32] focuses on the SOAP-based aspect of this interface where a developer may post a SOAP message with XML signatures. To exploit, all the attacker needs is a valid, signed SOAP message. Such messages can be easy to obtain; for example, developers tend to include them in public message postings in order to aid with debugging. The attacker may then use these messages and their keys to forge a new, malicious message. With this forged message the attacker can trick the host, Amazon, into thinking that the attacker is a legitimate administrative user for that domain, hence giving the attacker complete control over that domain. This attack was first described by McIntosh and Austel in 2005 [33].

It should be noted that Gruschka and Iacono [34] originally proposed a similar attack, though their version of the attack had the disadvantage of being time sensitive. However, Somorovsky et al. [32] later solved the time sensitivity problem, rendering the attack more feasible.

So far, we have encountered no research targeted at preventing these types of attacks–compromising and maliciously using the cloud management interfaces. While specific vulnerabilities can be addressed by software updates, such fixes can only be implemented when vulnerabilities are disclosed. Zero-day exploits are particularly dangerous in the context of cloud infrastructure compromise given the leverage an attacker can gain from a single successful attack. An open area of research, then, is how to harden or otherwise defend against attacks on previously unknown vulnerabilities in cloud management interfaces.

IX. Related Work

Several reviews have already been performed regarding both to the cloud infrastructure and its current state of security. Almorsy et al. [35] and break cloud infrastructure down into the component service models Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (See Section II). Chen et al. [36] go on to review whether or not there are any new threats or protections within cloud security. They make the point that some of the threats seen so far within the cloud infrastructure are new only in the sense that they are being seen in the cloud computing model rather being used to target single machines (e.g. installation of malware). Lastly, Lombardi et al. [37] give an overview of the current threat model of the cloud, providing both a list of attacks and the requirements for these attacks. Following their definition of the current threat model of the cloud, they present a detailed framework to categorize the attacks (our attack categorization is similar, but not identical, to theirs). While these reviews have made notable contributions to analyzing the current state of cloud security, none of them cover both cloud attacks and defenses.

X. Discussion

The fundamental issue with the move to the public cloud is that "hardware" is now much less trustworthy than before. Attackers in the cloud can run their code on the same hardware as the victim without bypassing any access controls; instead, they just need to manipulate the cloud provider so as to share resources with the target. This below-the-operating-system level of vulnerabilities is also strictly additive: all of the old vulnerabilities in operating systems kernels, system libraries, applications, and user behavior are still present.

The issues of co-located denial of service are being reasonably well addressed today, simply because this problem is fundamental to the business model of public cloud providers. If customers get poor service, they will take their business elsewhere. Co-located confidentiality breaches, particularly through cache attacks, as we have shown are being actively studied in the research literature. However, these attacks are all complex and are likely only to work against a small subset of virtual machine workloads where cryptographic operations take place very frequently.

While there is less work on cloud-specific data integrity, availability, and confidentiality issues, previous practice with non-virtualized resources can address many of these issues, at least partially. At the end of the day, though, placing data in long-term storage in the cloud is an act of trust. As we have shown, there are some technical solutions that can help reduce the amount of trust that must be placed in the cloud. In practice, however, these problems are more often being addressed through contracts and reputations. While such social arrangements might appear to be problematic, the scale at which cloud providers operate allows them to implement best practices for data management. As such, they may be more trustworthy than local storage for most customers.

Nevertheless, cloud providers are very tempting targets for attackers, particularly sophisticated ones. If they can get control of a cloud provider's infrastructure, whether through external or internal interfaces, they can control the fates of thousands of cloud customers and millions of individual users. The automated mechanisms that allow resources to be allocated and de-allocated on demand could become a huge force multiplier in the wrong hands. Research into how to harden this infrastructure is difficult as much of the current technology is proprietary and is controlled by relatively few companies. The cloud technology stack, however, is becoming more standardized with initiatives such as OpenStack [38]. No matter how hard it is to do, such research is needed so we can, at a bare minimum, better understand the risks we are running with the ongoing migration of our computational lives to the cloud.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST special publication 800-145, September 2011.

[2] Google Inc., "Google app engine – google developers," https://developers.google.com/ appengine/, accessed on March 24, 2013.

[3] Amazon Inc., "AWS elastic beanstalk," https://aws.amazon.com/elasticbeanstalk/, accessed on March 24, 2013.

[4] Dropbox, "Dropbox help - where does Dropbox store everyone's data? " https://www.dropbox.com/ help/7/en, accessed on March 15, 2013.

[5] J. Ciancutti, "Four Reasons We Choose Amazon's Cloud as Our Computing Platform," http://techblog.netflix.com/2010/12/four-reasons-we-choose-amazons-cloud-as.html, accessed on March 15, 2013.

[6] "Google site search solutions," http://www.google.com/enterprise/pdf/google_site_search_solutions.pdf, accessed on March 15, 2013.

[7] P. Peacock, "Web-based vs cloud-based," http://thecloudandme.com/2010/03/18/web-based-vs-cloud-based/, accessed on March 15, 2013.

[8] "What is (isn't) Google App Engine, https://developers.google.com/appengine/training/ intro/whatisgae, accessed on March 15, 2013.

[9] M. Christodorescu, R. Sailer, D. Schales, D. Sgandurra, and D. Zamboni, "Cloud security is not (just) virtualization security: a short paper," in Proceedings of the 2009 ACM workshop on Cloud computing security.

[10] A. Shieh, S. Kandula, A. Greenberg, and C. Kim, "Seawall: performance isolation for cloud datacenter networks," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010.

[11] D. Ongaro, A. L. Cox, and S. Rixner, "Scheduling I/O in virtual machine monitors," in Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, 2008.

[12] F. Zhou, M Goel, P. Desnoyers, R. Sundaram, "Scheduler Vulnerabilities and Coordinated Attacks in Cloud Computing," 10th IEEE International Symposium on Network Computing and Applications (NCA), 25-27 Aug. 2011.

[13] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan, "Volley: Automated data placement for geo-distributed cloud services," in Proceedings of the 7th USENIX conference on Networked systems design and implementation, 2010.

[14] Í. Goiri, F. Julià, J. Fitó, M. Macías, and J. Guitart, "Resource-level QoS metric for cpu-based guarantees in cloud providers," Proceedings of the 7th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON 2010). LNCS Vol. 6296, Springer, 2010.

[15] Amazon Inc., "Amazon EC2 SLA," https://aws.amazon.com/ec2-sla/, accessed on December 12, 2012.

[16] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in Proceedings of the 16th ACM conference on Computer and communications security, 2009.

[17] R. Owens and W. Wang, "Non-interactive os fingerprinting through memory de-duplication technique in virtual machines," in 2011 IEEE 30th International Performance Computing and Communications Conference (IPCCC), 2011.

[18] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," in Proceedings of the 2012 ACM conference on Computer and communications security, 2012.

[19] Y. Zhang, A. Juels, A. Oprea, and M. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," in 2011 IEEE Symposium on Security and Privacy (SP).

[20] J. Shi, X. Song, H. Chen, and B. Zang, "Limiting cache-based side-channel in multi-tenant cloud using dynamic page coloring," in 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), 2011.

[21] A. Srivastava, K. Singh, and J. Giffin, "Secure observation of kernel behavior," 2008.

[22] B. C. Vattikonda, S. Das, and H. Shacham, "Eliminating fine grained timers in Xen," in Proceedings of the 3rd ACM workshop on Cloud computing security workshop (CCSW '11). 2011.

[23] Amazon Inc., "Amazon customer agreement," https://aws.amazon.com/agreement/, accessed on December 12, 2012.

[24] K. Bowers, A. Juels, and A. Oprea, "HAIL: a high-availability and integrity layer for cloud storage," in Proc. of the 16th ACM conference on Computer and communications security, 2009.

[25] A. Juels and B. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proceedings of the 14th ACM conference on Computer and communications security, 2007.

[26] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proceedings of the 14th ACM conference on Computer and communications security, 2007.

[27] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010 Proceedings. IEEE, 2010.

[28] M. Raykova, H. Zhao, and S. Bellovin, "Privacy enhanced access control for outsourced data sharing," Financial Cryptography and Data Security, LNCS Vol. 7397, Springer, 2012.

[29] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in INFOCOM, 2010 Proceedings. IEEE, 2010.

[30] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proceedings of the 17th ACM conference on Computer and communications security (CCS '10), 2010.

[31] S. Gajek, M. Jensen, L. Liao, and J. Schwenk, "Analysis of signature wrapping attacks and countermeasures," in IEEE International Conference on Web Services (ICWS), 2009.

[32] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, and L. Lo Iacono, "All your clouds are belong to us: security analysis of cloud management interfaces," in Proceedings of the 3rd ACM workshop on Cloud computing security workshop, 2011.

[33] M. McIntosh and P. Austel, "XML signature element wrapping attacks and countermeasures," in Proceedings of the 2005 workshop on Secure web services. ACM, 2005.

[34] N. Gruschka and L. L. Iacono, "Vulnerable cloud: Soap message security validation revisited," in 2009 IEEE International Conference on Web Services (ICWS 2009).

[35] M. Almorsy, J. Grundy, and I. Müller, "An analysis of the cloud computing security problem," in Proc. of the 2010 Asia Pacific Cloud Workshop, Colocated with APSEC2010, Sydney, Australia, 2010.

[36] Y. Chen, V. Paxson, and R. Katz, "What's new about cloud computing security? " University of California, Berkeley Report No. UCB/EECS-2010-5, January 2010.

[37] F. Lombardi and R. Di Pietro, "Secure virtualization for cloud computing," Journal of Network and Computer Applications, vol. 34, no. 4, pp. 1113–1122, 2011.

[38] OpenStack Foundation, "Open stack," http://www.openstack.org/, accessed on March 25, 2013.