Strategies for Establishing a Root of Trust in a Cloud Environment

Ian T. Edington

Athabasca University

Abstract

Cloud platforms are more affordable, flexible and convenient than traditional computing platforms. However, large corporations are not taking advantage of them because the opacity of the service makes it hard to trust. For an entity to establish trust in a cloud environment the hardware and hypervisor layers of abstraction need to be made transparent and verifiable. Addressing this issue is complicated because many of the cloud providers benefits of flexibility and convenience are derived from the opacity of the platform. In this paper we explore the challenges of providing a trusted platform in a cloud environment. A concise problem definition is established based on enterprises needs and the limitations of cloud platforms. Current work is analyzed in terms of it's ability to offer guarantees of trust to cloud platform users. Finally, a solution is proposed that build on existing solutions to offer a model for trusted computing in the cloud.

Keywords: Root of trust, Cloud Computing, TPM, MEE

Strategies for Establishing a Root of Trust in a Cloud Environment

1. Introduction

The cloud has been a large topic of research since it's inception almost a decades ago. The benefits of cloud computing have been felt by many organization and have lead to the rise of many private and public clouds. AWS, Azure, GCE, and RackSpace have all built platforms for companies to build their data centers in the cloud.

With an ever increasing number of services and companies moving to a cloud architecture more stringent requirements are being set on cloud providers. Many organizations question who has access to their data, and wonder where their data is being stored. Although many organizations can accept these risks in return for the benefits of a cloud platform, some cannot. This group of organizations are unable to reap the benefits of cloud platforms because of a lack of enforcible trust.

This paper explores the cloud landscape, analyses current solutions, and expands on the current solutions to provide a more complete model of trusted cloud computing. In section 2 the background on the topic of security in the cloud is cover. In section 3 we look at challenges to providing the same security guarantees as in a private data center. In section 4 we identify and define the problem. In section 5 we analyze existing solutions. Finally, in section 6 we develop and propose a solution.

2. Background

Enterprise servers have evolved from large machines in private data centers to clusters of small machines in large data centers. Many factors have influenced this shift including the

introduction of Virtual Machines (VM), the decreasing cost of computing power, and increased security measures in system design. As this progression continued and distributed data centers became more ubiquitous, server rental became increasingly more common. Large companies were often more skilled at developing and managing data centers and if was often more cost effective for smaller companies to rent computation resources from larger companies than to create their own data centers. This trend has lead to the rise of cloud computing.

Like many new areas of study cloud computing has meant different things to different groups over the last decade. Much work has gone into defining what exactly constitutes cloud computing and that work can largely be summarized in NIST's definition of cloud computing:

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. (Mell and Grance 2011)

Virtualization was key in the growth of this new model of computation since it decouples the hardware from the operating system. OS startup time, portability, and reliability were all increased and could be done programmatically using Virtual Machine Managers (VMM). In this way the VMM was instrumental in allowing cloud architecture to advance as quickly as it has.

Running VMs in a cloud platform brings with it many benefits over buying and managing servers on premise. Servers can be provisioned on demand in a mater of minutes with minimal human interaction, rather than procuring, installing and provisioning bare metal servers, which often takes days. Removing a server is as easy as provisioning one. Scaling applications in

reaction or anticipation of user demand is possible because of the short startup time required by VMs. The peak usage of a data center is shared across multiple products and companies. When a single company required more compute power, it is rare that the data center will run out of resources, since they are pooled with a larger community. An application can be distributed to a larger geographic area at less expense than with traditional techniques. Outsourcing the physical and low level security of a cloud platform can often be advantageous to companies that lack the skill or resources of a large hosting company. All these benefits along with the flexibility of pay as you go billing with no capital investment has lead to massive growth in cloud platforms in every market segment.

However, moving from on premise servers to VMs in hosted environments introduced two layers of abstraction and potential security vulnerabilities along with those abstractions. With VMs a company still controls the operating system and software of a virtual machine, however, it does not control the hardware or the software platform. Although working with a trusted cloud provider is sufficient in most cases, having assurances of control is necessary for certain applications, especially regulated sectors like healthcare (Habib, Hauke, Ries, and Mühlhäuser, 2012). This exposes consumers of cloud products to three types of risks according to Ali, Khan, and Vasilakos; architectural, communication, and contractual (Ali, Khan, and Vasilakos 2015). Contractual risks have largely been solved by accreditation of the physical, hardware, and software solutions through third party audits. A large amount of research in the area of cloud computing has resulted in most of the technical challenges in Ali et al. 2015 paper being solved. Communication and architectural best practices have been developed and implemented in many cases, but they also rely on audits and accreditation. However, with the

speed of innovation and the rate at which vulnerabilities are found, annual audits are not enough to reasonably expect that a given sever will be functioning according to the audit. Take for example the Xen Project, which runs the larges cloud environments in the world including Amazons AWS (Xen Project 2017). The Xen Project suffered from 44 security advisories and provided 25 software releases on it's main product in the past year. A company relying on a yearly audit cannot guarantee that the software audited will be running on a given server at any given time.

As discussed in depth by Arthur and Challener (2015) this problem can be solved in a traditional on premise environment by employing a Trusted Platform Module (TPM) to establish a root of trust.

3. The Challenge in Providing a Cloud Platform Root of Trust

In general cloud platforms are not new technologies but are instead a reorganization of existing technologies in a new paradigm (Zhang, Cheng, and Boutaba 2010). In this way many of the challenges and solutions in providing a provably correct solution to this problem have come from other areas of computer science. Trusted computing platforms is a solved problem for on premise servers and has formed the basis for most of the attempted solutions to trusted computing in the cloud.

A root of trust provides a cryptographically signed chain from the first program booted on the system, up to the last piece of software configured in the chain. In order to ensure that the root of trust is not tampered with during boot, a hardware TPM is used to perform the cryptographic signatures and hold the resulting keys. The process is explained by Arthur and Challener:

> After reboot, a platform begins with trusted software called the core root of trust measurement (CRTM). The CRTM measures (calculate a digest of) the next software to be run and extends that digest into an even PCR. It then extends that software's configuration data into an odd PCR. This software, perhaps a BIOS, in turn measures and extends the next software, perhaps a master boot record. The measurement chain continues through the early OS kernel code and perhaps further. Security-critical configuration files are also measured. (Arthur and Challener, 2015)

By measuring each phase of the boot process a server can then be compared against a known state in order to prove that the system is running in the state it is mean to be running. In this way companies running their operating systems directly on hardware have provable measurements to show that the software is working as expected and establish a level of trust for that server. However, this currently does not work in cloud environment.

When a TPM is exposed to program level it provides that program with tools to monitor the system and continue the chain of trust. However, because these tools are hardware tools, they are server specific. There might be an arbitrary number of VMs on any given machine and VMs might travel from one machine to another. For this reason exposing the TPM to virtual machines could cause the TPM to react strangely when more than one VM uses it, and when a VM is migrated it will not have access to the TPM that is was signed with. Both of these issues stem from the limited number of Platform Configuration Registers (PCR) inherent in a hardware TPM, and would both cause catastrophic failure in a VM.

4. Problem Definition

The goal of this paper is to provide a Guest OS with a root of trust. A cloud environment inherently introduces risks into a computing environment. VMM vulnerabilities, and malicious hosts make it possible that the environment a customer is paying for isn't necessarily the one they have. The only known way to remedy this problem in on premise solutions is to use a TPM to establish a root of trust. In cloud providers root of trust analysis is complicated because of the dynamic nature of host OSs. Root of trust is often used within the host OS, but isn't passed into a guest OS. A solution is required for a guest OS to establish it's root of trust based on the host, with the following criteria:

- Host OS must be measurable by the TPM.
- Guest OS must have access to Host's TPM measurements.
- Guest OS must be measurable by the TPM.
- Guest OS must enable programs to verify both guest and host system.
- When a Guest OS is migrated it must not be allowed to execute code until the new Host OS is verified. When a Guest OS is started from a snapshot it must not be allowed to execute code until the new Host OS is verified.
- Must provide a solution without an exponential growth in TPM signatures (Jayaram 2014).

This solution would provide a root of trust through an entire VMs life cycle including, VM boot, migration, snapshot, rollback, and cloning. TPMs allow measurement of the Host, vTPMs allow measurement of the Guest, and solutions exist for reducing the exponential signature growth. The two issues that have not currently been addressed are Guest access to the

hardware TPM, and blocking code from executing on an unknown system. If such a solution existed, it would provide a root of trust for a Guest OS that could be re- evaluated if it were to move machines.

5. Related Work

This problem has been researched extensively both in trying to find a cloud root of trust and in terms of securing the VMM and guest VMs in other ways. Many of these solutions have crossover, and no one solution is feasible without one or more of the other solutions.

vTPM: a TPM within a VM

In order to solve the limitation of a physical TPM, Perez, Sailer, and van Doorn proposed a virtual TPM (vTPM), where one vTPM would be associated with each VM (2006). A vTPM is a virtualized software TPM that runs in the host VMM. It provides all the same interface as the TPM does in a hardware system, however, instead of executing in hardware it is virtualized by the host. Like a vCPU this has the benefit that it is not bound to any host and can travel with the VM to another host. Given a trusted platform, this solution allows a VM to create a root of trust for it's own operation and allows external moderation of the guest OS using it's private keys. With a trusted host this gives all the benefits of a TPM that a hardware system receives except that there is a layer between the hardware and the Guest Operating system that is not included in the root of trust. Because the VMM layer is no included in the root of trust this does not solve the problem of the trustworthiness of the underlying VMM and hardware. It is however the basis on which many other solutions have been built and would be required in any root of trust solution for a virtual machine.

Trusted Cloud Computing Platform

The paper from Santos, Gummadi, and Rodrigues expand on this concept in order to outline a potential solution for a Trusted Cloud Computing Platform (TCCP) (2009). Jayaram et al. build on the solutions of Santos (2009) and Perez (2006) to offer the most complete TCCP solution to date (2014). In this proposal the VMM would be a thin server, providing only the minimum features to allow for management of the VMs. In this solution the TPM is used to ensure that the TCCP is working correctly then passes the TPM results to a separate server that holds and allows inquiry's about a server. Using this model a VM would check the attestation server in order to verify that the machine it was running on was in good working condition before execution. During the regular life cycle of a VM, it will be moved from one machine to anther many times and the server it is running on will have patches applied to it. This model falls short when applied to the dynamic nature of a cloud platform. However, the strategy of using an attestation server has been used by many other solutions, such as release image signing, in order to guarantee uncontaminated software delivery.

Certified Virtualization Operating System

CertiKOS (Certified Kit Operating System) by Gu et al. uses a microkernel architecture for the host OS in order to separate VMM code from the underlying VMs (2011, 2016). The strategy moves many of the algorithms for virtual machine management from the kernel layer into the user layer code of the host OS. Only the minimum amount of functionality is required of the kernel to enforce protection and isolation. This allows the kernel to have a much smaller footprint and therefore is easier to certify correct. This greatly simplifies security and greatly decreases the amount of code needed to be signed by the TPM. The benefit to a TCCP, aside

from a provably correct kernel which is not covered in this paper, is a much smaller chain of trust.

6. Proposed Solution

The solution described here has both a software and a hardware component. A root of trust is only valid for a known server and since in a virtualized environment a server is only a temporary host the solution is portable.

Guests knowledge of virtualization

A core assumption of Virtual Machines is that a VM acts like it is running on a normal machine without knowledge of the virtualization. The main reason for this trait is to simplify the organization and execution of operating system in order for them to run both on hardware and in virtual machines. However, this trait goes against the nature of a trusted environment. In a trusted environment the layers higher up must have knowledge of the layers lower down in order to verify they are trustworthy. All of the current solutions still assume this trait, and require VMs in all to either stay on their machine of origin, or implicitly trust another machine in the cluster. Although it is possible to sign a vTPM using the original machines root of trust, both locking a VM to a machine, and transferring without a renegotiation of the root of trust, breaks the trust of the VM and should not be allowed. By continuing to enforce this separation of VM form hypervisor it makes the processes of establishing a root of trust more difficult.

If the guest OS was to be modified with knowledge of the underlying virtualization, it would be possible to pass a root of trust to the OS and have the guest OS validate it. In this model, a VMM would boot with a hardware root of trust, it would then start the VM with the vTPM's root of trust. However, instead of the guest OS having to implicitly trust the hosts root of

trust, it would be passed into the guest OS for verification. Before being trusted with execution the guest OS, the host OS would be subjected to a security audit by a third party in order to verify both the virtual root of trust and the hardware root of trust. Since hardware TPMs contain a unique certificates provided by their vendor, they are identifiable. Using a hardware TPM root of trust, a third party audit can verify the hardware, the VMM software that was booted, and the configurations for that server.

This is different than current solutions in that the process for trusting a server is separated from the process of trusting a running application. Separation of hardware and virtual verification results in a verification process that can be repeated by the VM at a different time with a different hardware root of trust. This addresses the issues of allowing the Guest OS have access to the hardware TPM measurements and enables the verification of both the guest and the host. However, it does not address the issue of blocking code from executing on an unknown system.

Currently, the TPM allows any program with access to add a layer to the root of trust. This is meant to allow a non owner to certify itself to a process higher up the root of trust. However, this means that anyone VM with access to the root of trust could sign itself, creating a potentially massive root of trust, which would be computationally inhibitive for a verification. This is a difficult problem that would need to be researched further because potential solutions require either modification to the TPM or implementing some kind of VM call to the host VMM.

Hardware CPU register

The second issue is guaranteeing that after a migration, rollback, or image clone, a VM is in-operable until the hardware root of trust has been verified. In order for this to work, some kind

of secret needs to be made available in a trusted system that can't be passed to a new system. By requiring this key in order to operate correctly, a CPU on a new system would be blocked from executing. Such a key exists due to advances in memory encryption. There exists an option for new processors to use a Memory Encryption Engine (MEE) described by Gueron (2016) to encrypt portions on the available memory. This solution would have three parts.

First, as part of the verification process the attestation server would return a key to the system that was verified. The key would be signed using the hardware TPMs public key. This would ensure that any use of the key could only be possible by the machine it was signed on. When a VM was moved to a new machine the key would have to be reissued.

Second, this key would need to be the primary encryption key for all Guest OS memory. Using the MEE the memory of a Guest OS could be encrypted using the key provided by the external source. Since the key is encrypted to only be used by the hosts TPM, any time the Guest OS moved to a new Host the key wouldn't work.

Third, the trap for an error in the MEE would need to be handled by an unencrypted portion of main memory that would resend for verification of the host. The attestation server would verify the new host TPM and the VM TPM and would deliver the same key except that it would be signed with the new hosts certificate. The key would be decrypted by the new host and the OS would continue where it left off.

Using these concepts together it is possible to develop an operating system that will not run on an untrusted system.

Difficulties with this implementation

There are considerable challenges to this proposed solution.

1. Integrating both a hardware TPM and a virtual TPM into an operating system is a major concern in the implementation of such a system. This communication is a challenge for both the VMM operating system as well as the OS running in the VM.
2. Rewriting an OS to take advantage of the MEE with an external key would be difficult but with the hardware support available would now be possible.
3. One of the benefits of VMs is how smoothly they move from one system to another. With this process special care will need to be made in the interrupt handling of a missed MEE key. The time it takes to re-verify the host is time that the VM will not be available.

Due to these restriction, computational overhead, and the difficulties of implementing such a system, only servers running sensitive payloads will implement this type of system.

7. Conclusion and Future Work

This study analyses current work in the field of trusted and transparent cloud platforms with the goal of implementing a trusted cloud platform that allows for all the benefits of cloud systems with the added benefit of a verifiable trusted platform. Current system do not allow for such a platform, however, advances in secure hardware in the form of Trusted Platform Modules and Memory Encryption Engines has made it possible for a solution to be developed.

Building on the work of Perez et al. (2006), Santos et al. (2009), Jayaram et al. (2014), and Gu et al. (2016), a solution is proposed to enable a verified and tamper resistant hardware root of trust that is flexible enough for a cloud environment. This solution uses a third party to verify the integrity of the host hardware and software separately from the guest VM. A key from the third party is used to encrypt the memory of the guest VM. When the guest VM is moved to

another machine, the key provided can no longer be used due to it being signed for the original host, which triggers a re-authentication of the new host.

This solution establishes a root of trust that can be verified by a third party, can be transported from one machine to another, and allows all the benefits a cloud provider enables.

An area that this paper does not address is the topic of secure deletion of a VM when it is moved, rolled back, deleted or cloned. Secure deletion is the process of overwriting the contents of a disk in order to destroy any data left on the disk. At this point an adequate solution for guaranteeing secure deletion is not in place. Since data privacy is the main concern of many enterprise customers waiting on trusted computing in the cloud, the process of provably deleting a VMs data after it is shutdown is an area that needs more research.

References

Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in cloud computing: Opportunities and challenges. *Information Sciences*, 305, 357-383.

Arthur, W., & Challener, D. (2015). A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security. *Apress*.

Gueron, S. (2016). A Memory Encryption Engine Suitable for General Purpose Processors. *IACR Cryptology ePrint Archive, 2016,* 204.

Gu, R., Shao, Z., Chen, H., Xiongnan (Newman) Wu, Kim, J., Sjöberg, V., & Costanzo, D. (2016, November). CertiKOS: An Extensible Architecture for Building Certified Concurrent OS Kernels. In *OSDI* (pp. 653-669).

Gu, L., Vaynberg, A., Ford, B., Shao, Z., & Costanzo, D. (2011, July). CertiKOS: a certified kernel for secure cloud computing. In *Proceedings of the Second Asia-Pacific Workshop on Systems* (p. 3). ACM.

Habib, S. M., Hauke, S., Ries, S., & Mühlhäuser, M. (2012). Trust as a facilitator in cloud computing: a survey. *Journal of Cloud Computing: Advances, Systems and Applications, 1(1)*, 19.

Jayaram, K. R., Safford, D., Sharma, U., Naik, V., Pendarakis, D., & Tao, S. (2014, December). Trustworthy geographically fenced hybrid clouds. In *Proceedings of the 15th international middleware conference* (pp. 37-48). ACM.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

Perez, R., Sailer, R., & van Doorn, L. (2006, July). vTPM: virtualizing the trusted platform module. In *Proc. 15th Conf. on USENIX Security Symposium* (pp. 305-320).

Santos, N., Gummadi, K. P., & Rodrigues, R. (2009). Towards Trusted Cloud Computing. *HotCloud, 9(9)*, 3.

Xen Project. (2017) xen-project/xen Releases. Retrieved July 25, 2017, from https://github.com/xen- project/xen/releases.

Xen Project. (2017) Advisories, publicly released or pre-released. Retrieved July 25, 2017, from https://xenbits.xen.org/xsa.

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications, 1(1)*, 7-18.