SCIENCE CHINA
Information Sciences

• INSIGHT •
Special Focus on Advanced Technology of Software
May 2016, Vol. 59 050105:1–050105:3
doi: 10.1007/s11432-016-5549-1

Theory and methodology of research on cloud security

Hai JIN*, Weiqi DAI & Deqing ZOU

Services Computing Technology and System Lab, Big Data Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Received November 15, 2015; accepted January 16, 2016; published online April 8, 2016

Citation Jin H, Dai W Q, Zou D Q. Theory and methodology of research on cloud security. Sci China Inf Sci, 2016, 59(5): 050105, doi: 10.1007/s11432-016-5549-1

Cloud computing is a model for enabling ubiquitous, convenient, on-demand access to a shared pool of configurable computing resources. According to Gartner's survey, cloud computing has been at the head of a list of the top ten IT strategic technologies for some time. Cloud computing is currently undergoing vigorous development and promotion by many leading Chinese and foreign enterprises, and it has become the basis for a new generation of IT industries. According to Gartner, the greatest obstacles to the further development of cloud computing are posed by security and privacy issues. Cloud security incidents have occurred frequently in recent years. Therefore, it is an extremely urgent matter to find a solution to cloud security issues.

To secure the cloud architecture, we need to overcome three main challenges, namely, the need for a secure execution environment for cloud virtual machines (VMs), the problem of time and status inconsistency of VMs, and the need for high reliability in the uncertain cloud. Challenge I: Current VM-based cloud computing practice does not offer an efficient security execution environment for sensitive applications on cloud-end computers. Challenge II: Although the virtualization feature of VMs enhances dynamism in the cloud, it also has an impact on state consistency among cloud services, which gives rise to fatal security issues. Challenge III: Enhancing the reliability of cloud services through approaches such as fault diagnosis and dynamic software updating, when carried out in traditional ways, consumes much time and resources [1].

To solve Challenge I, we propose a trusted execution environment (TEE) [2] solution to allow multiple customers or VMs on a commodity cloud-end platform to simultaneously enjoy dynamic root of trust for measurement (DRTM), as in a secure execution environment, without requiring expensive extra hardware. A TEE can support a spectrum of application needs, ranging from pure cryptographic libraries to fully fledged trustworthy software. Moreover, it can provide a secure execution environment for assured digital signing [3] which enhances data trustworthiness for a VM. The resulting system architecture consists of two components: the vDRTM and the TEE. From a technical perspective, the novel aspect of our work lies in the virtualization of DRTM. The vDRTM has three main components: the vDRTM-enabled vTPM Manager (in the vTPM domain), the vD-CRTM (in the virtual machine monitor, VMM) and the Virtual LPC. The vDRTM-enabled vTPM Manager supports control of locality, which is used to determine the origin of trusted platform module

* Corresponding author (email: hjin@hust.edu.cn)
The authors declare that they have no conflict of interest.

(TPM) commands (because in the Xen hypervisor [4] that we have employed in our experiments, only locality 0 is used), and is obtained by modifying the Xen vTPM Manager. The vD-CRTM can provide each VM residing on the same physical machine with a DRTM-like trusted execution environment. The Virtual LPC is the virtual bus for communication between the vD-CRTM and the vTPM Manager, and mimics the low-pin-count (LPC) bus between the CPU and the TPM in a nonvirtual system. The TEE component provides a trusted execution environment in which a sensitive application runs on top of the TEE kernel at any time when requested by the VM, and provides a trusted measurement to the third party.

To solve Challenge II, we propose a rollback-resilient vTPM (rvTPM) system and TPM for the cloud. (i) Rollback resilience of the vTPM solves the problem of time-related inconsistency of VMs in the cloud by analyzing particular vulnerabilities inherent to the current design of VM rollback mechanisms, such as attack recovery. The VM rollback mechanism can be exploited to launch attacks against cloud environments [5] with or without the TPM. Operation of a VM without (v)TPM rollback can lead to loss of an application's state when this state is rolled back to an earlier one. The basis of the abovementioned attack against systems that utilize a (v)TPM is VM-vTPM discrepancy. It is not secure to rollback the entire vTPM instance, because the monotonicity of the vTPM counter is then broken, resulting in vulnerability to replay attacks. For example, an application may lose monotonicity after such a rollback operation, allowing a brute-force attack or a replay attack to be mounted. We propose rolling back only those platform configuration registers (PCRs) that measure the integrity of the VM and those that are used in TPM functions. Rolling back the key hierarchy in system-persistent storage (managed by the Trusted Computing Group (TCG) software stack, TSS) is not appropriate, because this may lead to loss of the registered keys. Moreover, the TSS key handles of unregistered keys, which are compromised or expired, may reappear after a rollback operation. All vTPM instance handles must be flushed after each rollback operation, because they can be abused to launch attacks. To allow the application or other VMs to be aware of the rollback event of a VM, we add an extra group of PCRs. The rvTPM protects the VM against attacks that attempt to cause loss of an application's state or to cause VM-vTPM discrepancy. (ii) Since VMs are unlikely to be aware of dynamic changes in other VMs, such as a VM joining or leaving a virtual machine group (VMG) and VM migration, it is difficult to maintain a consistent status of cloud services. We present a design of TPM for the cloud (TPMc) [6] that is a system that manages and attests the security state of VMGs. TPMc is a vTPM-based system for monitoring a group of VMs, and is designed to deal with new challenges to which a vTPM is vulnerable. First, there are 24 extra PCRs to provide meaningful remote attestation to verify the whole VMG at one time. Furthermore, TPMc provides a global PCR view of all the vTPM to allow the verifier to monitor the PCRs states of all the VMs in the VMG. In addition, all the TPMc instances in different nodes should work consistently as one single instance, and the nonvolatile storage should be shared safely among the TPMc instances in the same VMG. Moreover, TPMc provides a sharing key hierarchy mechanism to keep the overheads (e.g., the overhead of key exchange) of communication at a low level. Finally, we have designed a new seal & unseal function and a new data-sharing protocol to simplify the data-sharing procedure between VMs and improve the security of keys in the vTPM.

To solve Challenge III and achieve high reliability of cloud services, we focus on software updating. We present two methods, online and offline, to update software. (i) To update the software of cloud services online, we present Replus [7], a new dynamic software updating system that balances practicality and functionality. Replus aims to retain backward binary compatibility and support for multithreaded programs. By separating the duties of developers and customers, Replus allows customers without developer-level software knowledge to update software. More importantly, in the absence of specific compiler support, Replus can now patch programs that are difficult to update at runtime, as well as programs that may incur an indefinite delay (the time between the start and finish of an update) in dynamic software updating. The key technique used in our solution is to update the stack elements for the new version of the program using two new mechanisms: immediate stack updating, which immediately updates the stack of a thread, and timely stack updating, which updates only the stack frames of the to-be-updated functions without affecting others. Replus also adopts an instruction-level updating mechanism, which is more efficient for specific security patches. (ii) To update the software of cloud services offline, we present a new software update [8] model called update as a service (UaaS) to handle VM offline updating automatically. Multiple VMs share a single software update service, and multiple update strategies are provided for single pieces of software, which can be customized at any time. We

first propose the software update as a cloud service for cloud users, which can be used to update the software for offline VM images. UaaS is based on a low-overhead software information collection method and a fast VMs-to-be-updated search algorithm to make the service more efficient.

Acknowledgements This work was supported by National Basic Research Program of China (973) (Grant No. 2014CB340600) and National Natural Science Foundation of China (Grant No. 61272072).

References

1 Tsai W T, Bai X Y, Huang Y. Software-as-a-service (SaaS): perspectives and challenges. Sci China Inf Sci, 2014, 57: 051101
2 Dai W Q, Jin H, Zou D Q, et al. TEE: a virtual DRTM based execution environment for secure cloud-end computing. Future Gener Comput Syst, 2015, 49: 47–57
3 Dai W Q, Paul P T, Jin H, et al. Enhancing data trustworthiness via assured digital signing. IEEE Trans Depend Secure Comput (TDSC), 2012, 9: 838–851
4 Shi L, Zou D Q, Jin H. Xen Virtualization Technology (in Chinese). Wuhan: Huazhong University of Science and Technology Press, 2009
5 Jin H. The Virtualization of Computing System Principles and Applications (in Chinese). Beijing: Tsinghua University Press, 2008
6 Zou D Q, Qiang W Z, Jin H. The Principles and Application of Trusted Computing Technology (in Chinese). Beijing: Science Press, 2011
7 Chen G, Jin H, Zou D Q, et al. A framework for practical dynamic software updating. IEEE Trans Parallel Distrib Syst, 2016, 27: 941–950
8 Liu K, Zou D Q, Jin H. UaaS: software update as a service for the IaaS cloud. In: Proceedings of IEEE International Conference on Services Computing, Chicago, 2015. 483–490