POLITECNICO
DI TORINO

Dipartimento di
Automatica e Informatica

PhD in Computer and Control Engineering

XXVIII cycle

Advisor

Prof. Antonio Lioy

Distributed System Security

Enable trust with remote party

PhD Candidate: Tao Su

Introduction

In distributed systems, communication channels can be well protected by secure protocols like TLS or IPsec, but there is no guarantee that the secure channel end points will behave as agreed (because of customised services or remote attacks).

Software-based security solutions are no longer sufficient to deal with the current scenario (because the device is on-line and remote), but must be coupled with stronger means such as hardware-assisted protection.

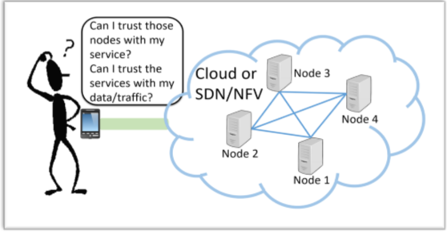

Can I trust those nodes with my service?
Can I trust the services with my data/traffic?
?
Cloud or SDN/NFV
Node 1
Node 2
Node 3
Node 4

Figure 1: Trust problem in distributed systems.

The objective of my research is to evaluate the integrity state of remote platforms and services running in them based on hardware-based evidence. The evidence needs to prove the platform is booted into a trust state by loading trusted components in a predefined order and the services running in the platform are loaded with trusted executables and configurations.

Idea - Remote Attestation

If the first execution step can be trusted and each step correctly measures the next executable software (i.e. computing its digest), then the overall system integrity can be evaluated in a reliable manner.

Pre-requisites: isolated storage and unique identifier – provided by *Trusted Platform Module* (TPM), a separated hardware chip.

Each TPM has 24 *Platform Configuration Registers* (PCRs) which only accept *extend* operation to store the measures in following way:

$$PCR_{new} = HASH(PCR_{old}||measure)$$

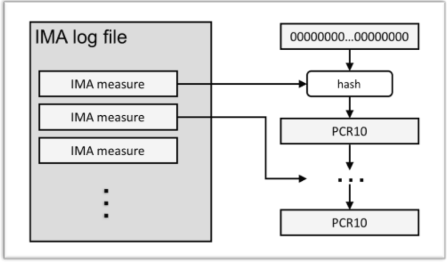


Figure 2: Extend operation.

Once received a RA request, the TPM creates a signature of the received NONCE (generated by the requestor/verifier) and the current PCRs, with a key which never leaves the TPM (this is called *quote).*

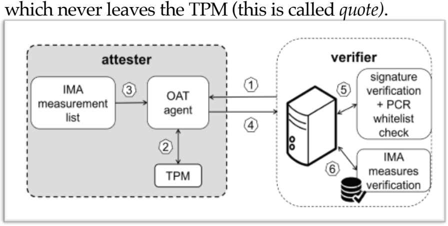

attester
IMA measurement list
3
OAT agent
2
TPM
1
4
verifier
5
signature verification + PCR whitelist check
6
IMA measures verification

Figure 3: Operations in the RA framework.

Remote Attestation Framework

The proposed remote attestation solution [2] is the first complete solution to cover the platform booting phase and service loading phase (Fig.3). Moreover, the associated implementation is flexible that can be used for both physical nodes and virtual machines. Additionally, it can be adapted to improve other security techniques, e.g. trusted channel.

Boot integrity: the components loaded in boot phase are extended in different PCRs.

Service integrity: Linux Integrity Measurement Architecture (IMA) measures executables and configuration files, and extends the measures into PCR 10 before they are loaded into kernel memory (Fig.2).

Virtual Container Attestation

Virtual container provides more lightweight virtualisation environment than the conventional virtual machines, which is crucial for SDN/NFV environments. The goal of this work is to attest the services running inside virtual containers in a trusted manner. Thus to identify the compromised container and restore the trusted state of the platform without resetting it (Fig. 4). This result of this work has been submitted to Computers & Security [3].

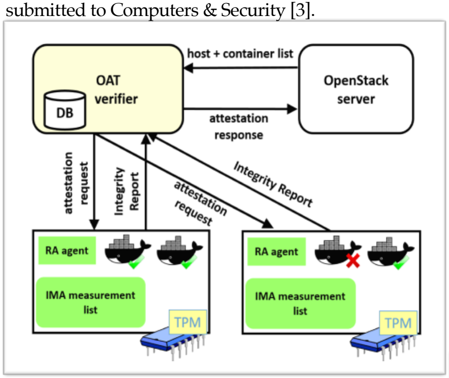
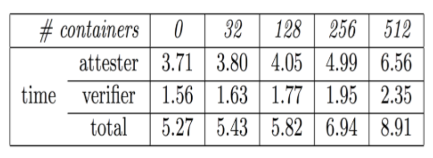


Figure 4: RA architecture for Docker containers.

Experiment Results



	# containers	0	32	128	256	512
time	attester	3.71	3.80	4.05	4.99	6.56
	verifier	1.56	1.63	1.77	1.95	2.35
	total	5.27	5.43	5.82	6.94	8.91

Figure 5: Overall RA performance (in second).

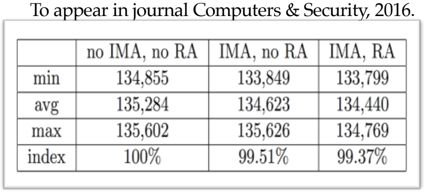
Research Outcome

My research outcome is twofold. First it offers a complete remote attestation solution and the associated implementation, with the integrity of not only the attesting platform, but also the services running in it evaluated based on hardware-based evidence. Moreover, by extending this solution, services running inside virtual containers can also be attested in trusted manner. Both achievements are relevant to improve distributed systems security, especially in SDN/NFV environments [1].

Regarding to future work, the performance of the framework can be further improved by using firmware TPM instead of physical TPM. Second, the framework should be merged with existing management and orchestration entity (e.g. OpenMANO).

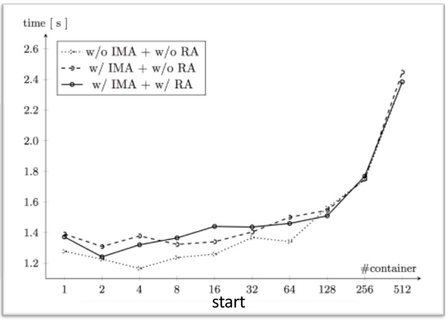
References

1. Lioy A., Su T., et al. Trust in SDN/NFV environments. To appear in book Guide to Security in SDN and NFV - Challenges, Opportunities, and Applications, 2016
2. Su T., et al. Trusted Computing Technology and Proposals for Resolving Cloud Computing Security Problems. Published in the book Cloud Computing Security: Foundations and Challenges, August 2016, pp. 345 - 358.
3. Su T., et al. Practical integrity verification for the Docker lightweight virtualization environment. To appear in journal Computers & Security, 2016.

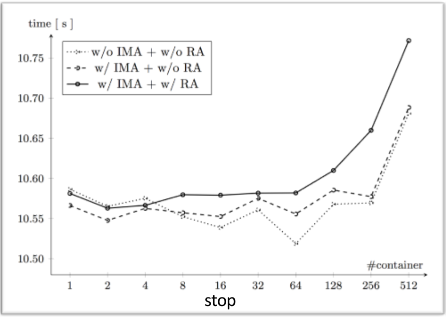


	no IMA, no RA	IMA, no RA	IMA, RA
min	134,855	133,849	133,799
avg	135,284	134,623	134,440
max	135,602	135,626	134,769
index	100%	99.51%	99.37%

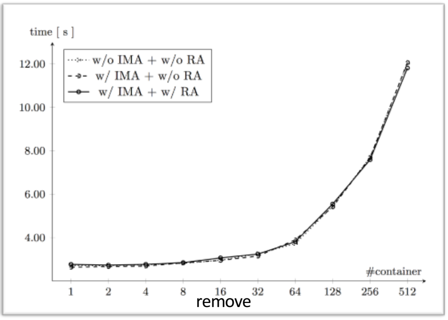
Figure 6: Performance impact with iterated hash operations.



start



stop



remove

Figure 7: Performance impact to basic Docker container operations.