POLITECNICO DI TORINO

Dipartimento di Automatica e Informatica

PhD in Computer and Control Engineering

XXXI cycle

Supervisor

prof. Antonio Lioy

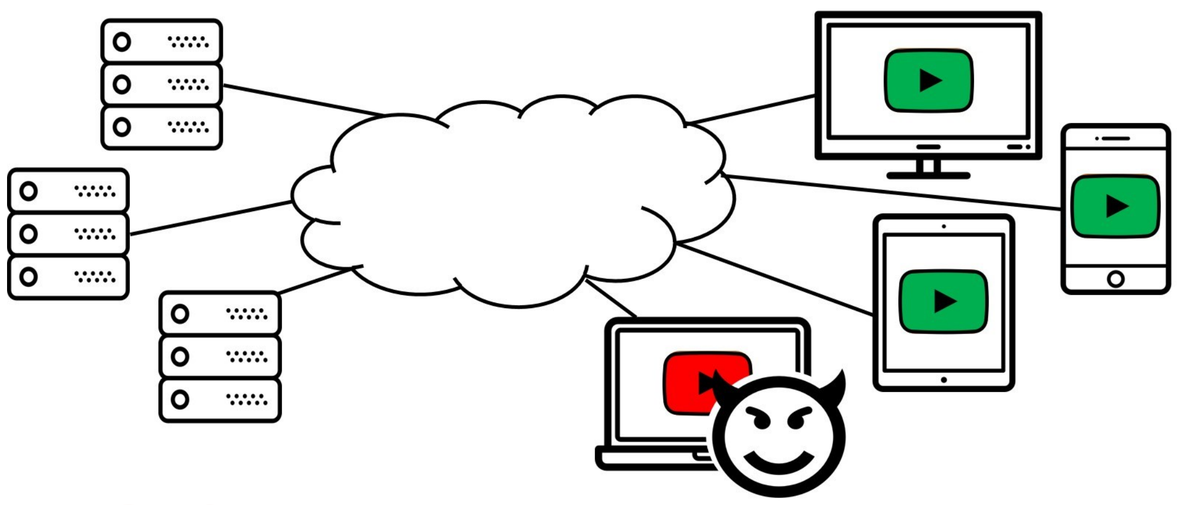
Software Attestation with Static and Dynamic Techniques

PhD Candidate: Alessio Viticchié

1. Introduction

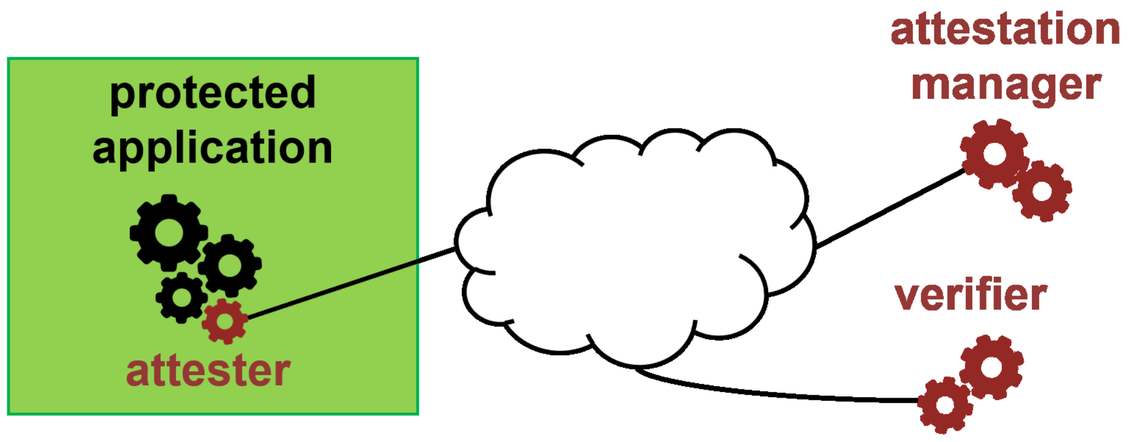
Modern systems are highly distributed and the end devices are very diverse.

Man-At-The-End attacks (MATE): attackers tamper with software applications in contexts where they have full privileges. Hence, hardware independent protections for software execution correctness are needed.



2. Software attestation

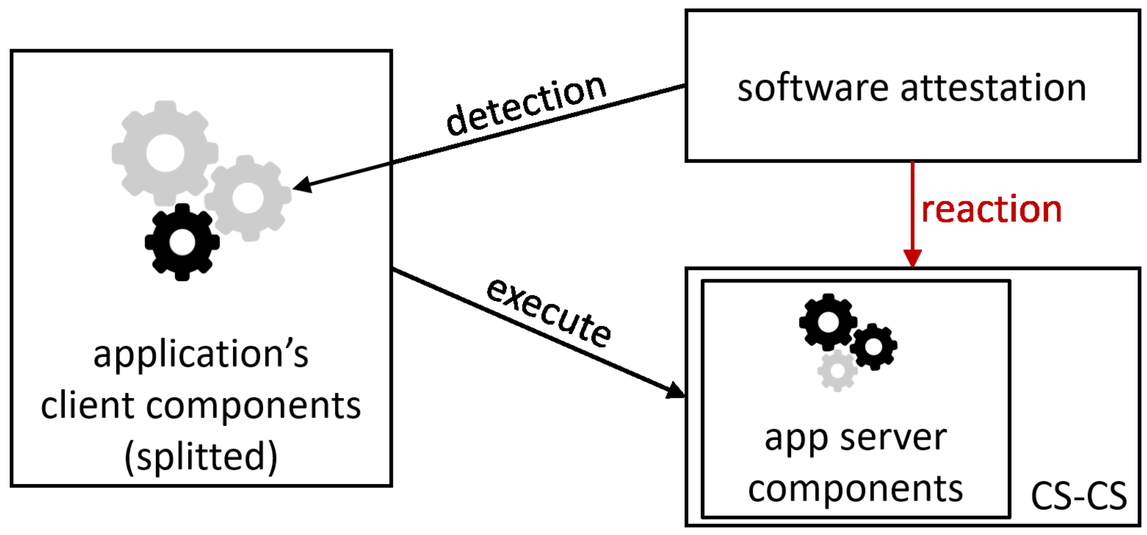
Software attestation is a software-only, remote, integrity checking technique. The *Attestation Manager* triggers the attestation procedure. The *Attestation Manager* sends an attestation request. The *Attester* extracts integrity evidences and sends them back to the server side. The *Verifier* checks what received to prove target integrity.



3. Static Software Attestation

The classical way, it monitors the integrity of program's binary in memory. Attester collects bytes from memory via a random walk and sends a digest of them to the verifier that compares it with a precomputed one.

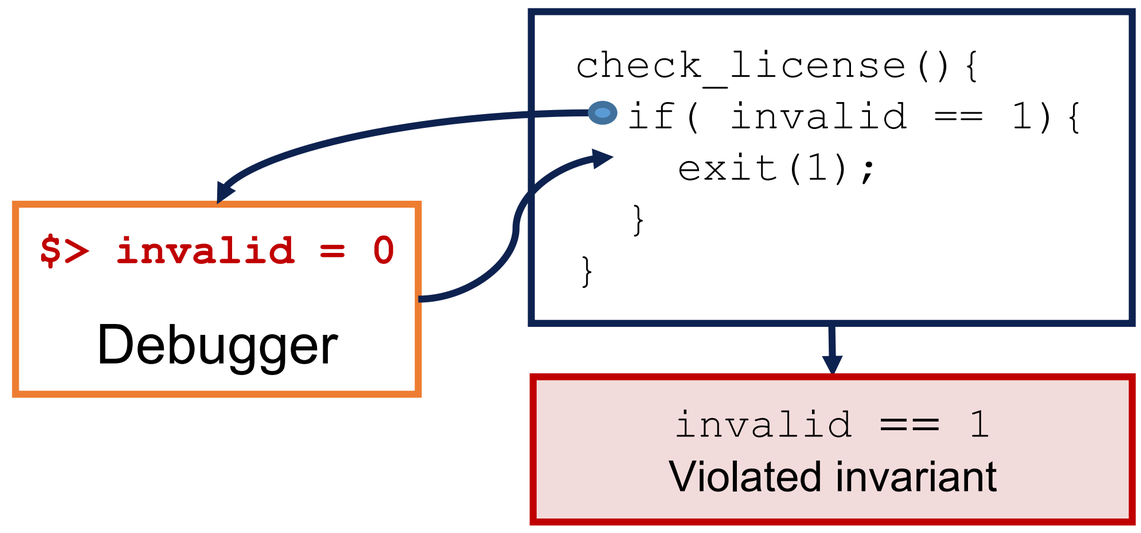
Effective but no reaction to tamper. Then, *Reactive Attestation*: robust detection-reaction technique given by coupling of static software attestation and *Client-Server Code-Splitting (CS-CS)* [1].



4. Dynamic Software Attestation

Invariants monitoring (IM): to monitor integrity based on dynamic properties, i.e. likely invariants that are logic assertion valid for a portion or for the whole program.

Idea: use likely invariants to model software behavior thus to check execution integrity.



Good but not working in practice, IM suffers from false positives and, much worse, from false negatives. Those issues cannot be overcome then the technique is not useful in practice [2].

5. Conclusions

Software attestation is not sufficient to exhaustively protect a program. Static software attestation only detects very specific attacks. Invariants Monitoring is not useful for security purposes. For the future, robust software analysis techniques are needed to better model software behaviour.

6. References

1. Viticchié A. *et al*., "Reactive Attestation: Automatic Detection and Reaction to Software Tampering Attacks", SPRO 2016, DOI: 10.1145/2995306.2995315
2. Viticchié A. *et al.*, "On the impossibility of effectively using likely-invariants for software attestation purposes", ISYOU June 2018, DOI: 10.22667/JOWUA.2018.06.30.001