CERIAS

The Center for Education and Research in Information Assurance and Security

PURDUE UNIVERSITY

Security Auditing in Service Oriented Architecture

M. Azarmi , P. Angin, B. Bhargava, N. Ahmed , A. Sinclair

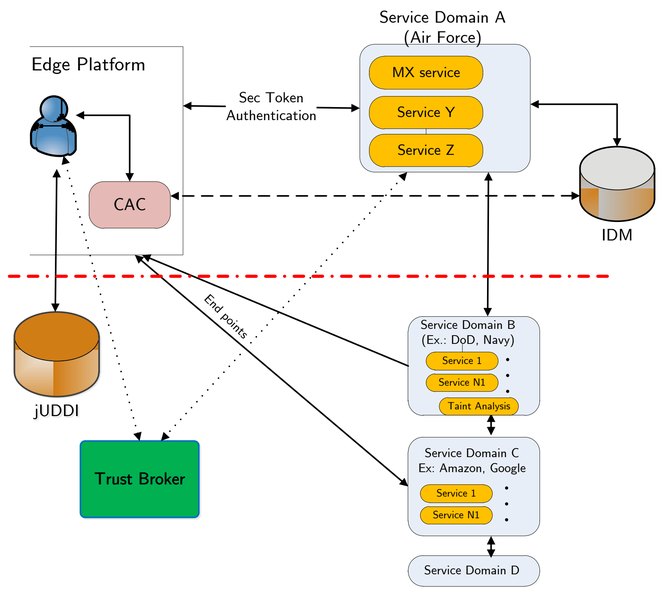
Abstract

Service oriented architectures (SOA) introduce new security challenges not present in the single-hop client-server architectures due to the involvement in multiple service providers in a service request. Considering the additional security threats on SOAs, the interactions of independent service domains could violate service policies or SLAs. We provide an efficient solution for auditing service invocations in SOA. This solution is transparent to the services, which allows using legacy services without modification. Moreover, we are investigating the transition to the cloud computing.

Challenges in SOA Security

Security Auditing in SOA (Problem Statement)

The end-to-end SOA infrastructure consists of a client making a request to the initial trusted services/domain and that service can make a service call to another service from a trusted domain or an untrusted public domain.



- Authentication and authorization does not take place across intended end points, such as between the requestor and service provider.
- Termination at intermediate steps of service execution exposes messages to hostile threats, for example, Man-in-the-Middle attacks.
- External services are not verified or validated dynamically (uninformed selection of services by user).
- User has no control on external service invocation within an orchestration or through a service in another service domain.
- Violations and malicious activities in a trusted service domain remain undetected.

Contributions

- A transparent service invocation control mechanism for SOA using service-level dynamic taint analysis.
- A trust broker (TB) system that maintains information about trustworthiness of services and categorizes them. TB is used for dynamic validation and verification of services and keeps track of history of service invocations.
- A secure end-to-end message origin authentication for Web service client requests and Web service providers to ensure confidentiality and integrity—even in the presence of man-in-the-middle attacks. This solution is based on the common access card (CAC).

Trust Broker Subsystem

Major functionality of TrustBroker module:
- Maintaining a list of certified services
- Evaluating the trust level of a given service
- Maintaining an end-to-end session of service invocation

Trust calculation: The trust value T_s for a service S, with SLA trust value L, getting feedback F at time t is updated using the equation:

$$T_s(t) = \beta \times [\alpha \times T_s(t-1) + (1-\alpha) \times F] + (1-\beta) \times L \text{ ,where } \alpha < 0.5$$

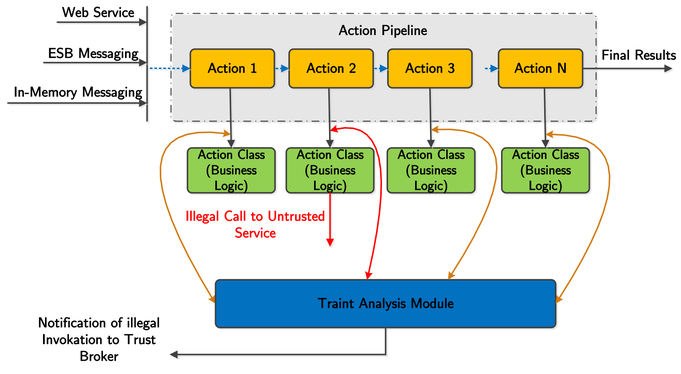
- ➡ β: the weight for the properties of the service such as its compliance with WS* standards
- ➡ α: past reputation of the service
- ➡ *F:* feedback parameter

Taint Analysis Subsystem

The *taint analysis* (TA) module monitors the activity of services and inspects the data exchanges (information flow) between them to detect certain events. Monitoring is done mainly for two activities; one is to check the compliance of those domains (or services) to their registered SLA agreement, and the second is the consumption of their data into the trusted services domain.

- Implemented using AOP (Aspect Oriented Programming)
  - ➡ Defining a set of *pointcuts* to intercept communication invocations.
  - ➡ Flexible to modify the level and granularity of interception based on audit policies.
  - ➡ Dynamic/load-time instrumentation of *class-file* in the SOA action pipeline (oblivious to the application server)
  - ➡ No access to source code or change in services needed.

Taint Analysis feedback will be used for:
- Monitoring information flow policies and tracking violations in a real-time SOA environment and logging for later reporting to clients.
- Decreasing the trust level of a service if it passes a message to a suspicious service.
- Adjusting trust levels for use in secure service composition.



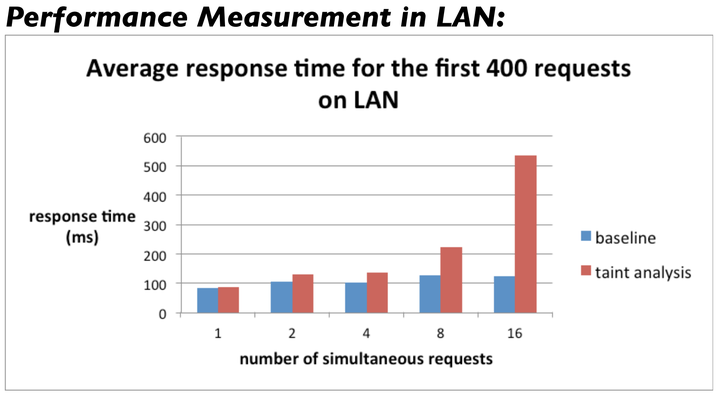
Interaction of TA module with SOA action pipeline

Experiments and Transition to Cloud

Implementation:
- Prototype based on JBoss ESB and JBoss AOP
- Three services and jUDDI registry

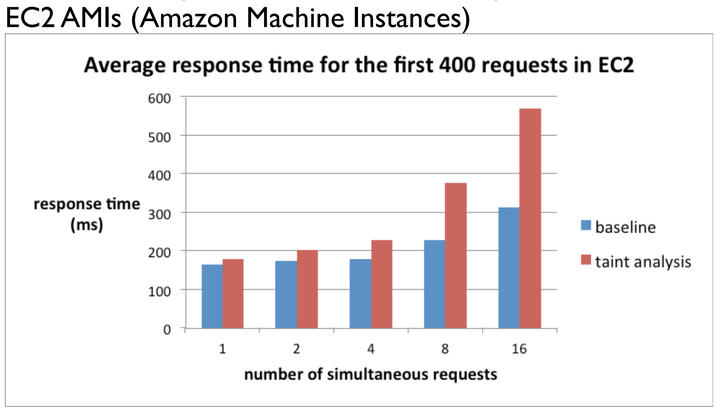
Security Evaluation:
The implemented prototype was evaluated in terms of its effectiveness in mitigating XML rewriting attacks and denial of service (DoS) attacks.

Performance Measurement in LAN:



Experiments in the Amazon EC2 Cloud:
- Amazon EC2 Cloud was used to study the impact of migration of the proposed end-to-end security solution to the Cloud.
- Multi-tenancy
  - ➡Sharing resources hampers resource availability and performance estimation
  - ➡Could pose additional security threats
- Instance IP is not public, we need to bind Elastic IP with an instance for public access (Solution: Hard-coded EIP in source code)

- We have conducted experiments to measure the performance impact of using cloud computing.
- In this setting, service domains were placed in Amazon EC2 AMIs (Amazon Machine Instances)



Future Work

- Investigating new threats for SOA-based systems in cloud computing environments
- Offloading the non-security-critical computation to the cloud
- Using TPMs along with taint analysis framework to provide a stronger security

Acknowledgement

We thank Dr. Xiangyu Zhang, Rohit Ranchal, Bala Gnanasekaran, Guneshi Wickramaarachchi, Lotfi Ben Othmane, Nwokedi Idika for their contributions.

CERIAS

Discovery Park e-Enterprise Center