Recovery from Security Intrusions in Cloud Computing

Dário Nascimento

68210 - MERC
Instituto Superior Técnico
Universidade de Lisboa

Supervisor:

Prof. Miguel Pupo Correia

Agenda

Motivation

Related Work

Proposed Solution

- Goals
- Architecture
- Evaluation

Conclusion

- Schedule
- Conclusion

Motivation

Increasing number of critical applications in Cloud

Intent to compromise:

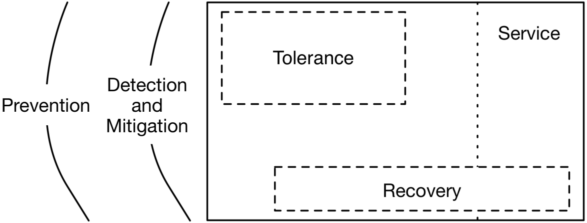
- Confidentiality
- Integrity
- Availability

Intrusion:

- Intentional vulnerability exploitation
- Malicious fault

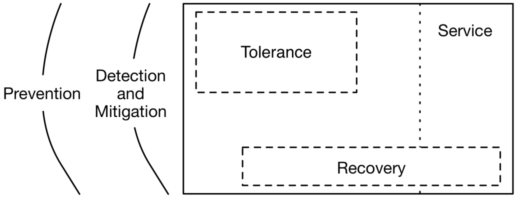
Recover the application **integrity** to prevent **losses**

Motivation


Prevention
Detection and Mitigation
Tolerance
Service
Recovery

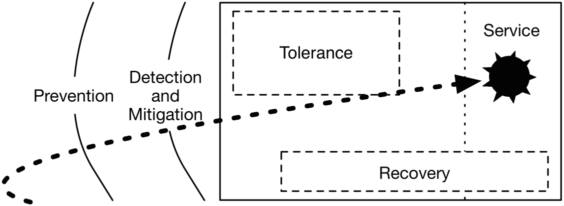
Motivation

- Software flaws
- New attack methods
- Configuration and usage mistakes (malicious or accidental)
- Legitimate requests



Motivation

Intrusions and failures happen!



Problem statement

How to recover from intrusions in PaaS?

Goals

Accept intrusions and remove their effects

- Identify the intrusion effects
- Remove intrusion effects
- Recover the application integrity
- Tolerate intrusions: recovery without exposing downtime [1]
- Recover from user and administrator mistakes

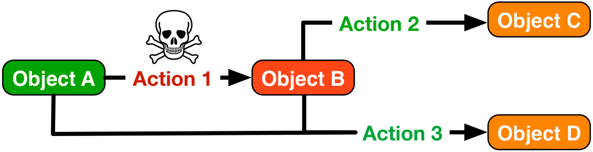
[1] Does not replace the prevention and tolerance

1. Identification of intrusion effects

Goal: Identify the intrusion actions or objects

- IDS: *[Taser,ITDB,Phoenix,Retro,Dare,Goel et al., Undo for Operators]*
- Software update *[Warp,Aire]*

1. Identification of intrusion effects

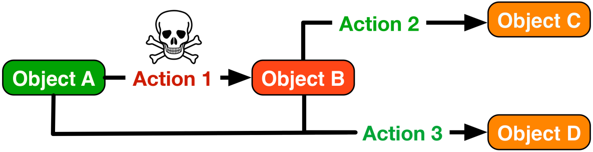


2. Remove intrusion effects

- Versioning *[Phoenix, Warp,Aire]*
- Snapshot *[Taser, Retro,Date, Undo for Operators]*
- Compensation *[Goel et al,ITDB]*

storage vs computing

3. Recover the application integrity



- No replay *[Taser, ITDB, Phoenix]*
- Taint via replay *[Retro,Dare,Goel et al,Warp,Aire]*
- Replay all *[Undo for Operators]*

Recovery: Where?

- Operating system *[Taser,Retro]*:
  - System calls, files and sockets
- Database *[ITDB,Phoenix]*:
  - Read and write sets: table, table block, row or field
- Web Applications*[Goel et al, Warp,Aire,Undo for Operators]*:
  - User requests and database transactions

Problems

- **Scalability**
  - Single database and server
- **Integration**
  - Lack of generic application support
  - Configuration per application
- **Application downtime**

Project Goals

Shuttle: Intrusion recovery service for PaaS

- PaaS Integration
  - Standard architecture for Web Applications
  - Service-oriented database access through provided libraries
  - Service available without setup and configuration
- NoSQL databases

Project Goals

Remove the effects of:

- Software flaws
- Corrupted requests and data
- Intrusions in PaaS instances

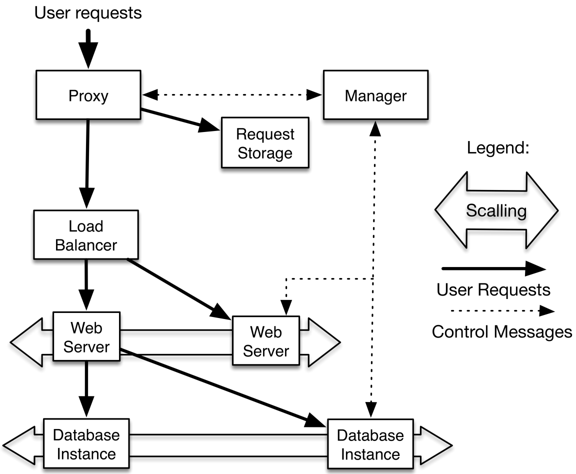
Project Goals

- Support software updates
- Low runtime overhead
- NoSQL database snapshot
- Recover without stopping the application

Architecture: Overview

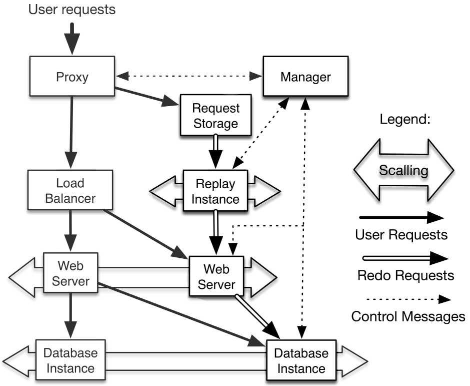
1. Records the user requests
2. Tracks the dependencies between requests using the database
3. Loads a snapshot
4. Replays the legitimate user requests

Architecture: Recording



Architecture: Recovery

1. Load a previous snapshot in background
2. Get the requests order using the graph
3. Send the requests in parallel using the replay nodes

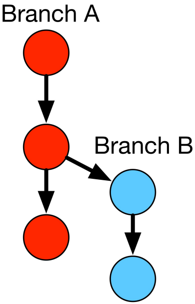


Architecture: PaaS

- PaaS controller lunches new database and application instances
- Clean images: replace corrupted instances
- Pay-per-use model
- Virtually unlimited computing and storage resources

Architecture: Branching

- Support multiple recovery branches
- The branch is defined by the request header



Evaluation

- **Prototype:** Java Servlet (Spring Framework) version of Question & Answers System
- **Data:** Data crawled from Stackoverflow.com
- **Database:** Cassandra and Voldemort (Key-Value store, DynamoDB)
- **PaaS:** OpenShift, AppScale (Google App Engine)
- **IaaS:** OpenStack and Amazon Web Services or Google Cloud Platform

Evaluation

- **Record impact:** Delay, throughput, resource usage, maximum load
- **Replay:** Precision, recall, duration and scalability
- **Integrity and Availability:** Corrupted and unavailable data during recovery
- **Concurrency:** Correctness and performance improvement
- **Cost:** Monetary cost in a public cloud provider

Schedule

ACTIVITY	PLAN START	PLAN DURATION	PERCENT COMPLETE	1	2	3	4	5	6	7	8	9
								Months				
Preparation												
Related Work Reading	1	4	**100%**	■	■	■	■					
PaaS and Database options	1	2	**100%**	■	■							
Architecture	3	2	**100%**			■	■					
Implementation												
Testing tool	2	1	**100%**		■							
Q&A application	3	1	**100%**			■						
Proxy	2	2	**100%**		■	■						
Redo Instance	4	2	**100%**				■	■				
Database Instance	3	3	**100%**			■	■	■				
Manager	4	2	**100%**				■	■				
Integration with PaaS	6	1	**50%**						■			
Testing												
Validation (Local)	7	1	**0%**							▨		
Evaluation (Cloud)	7	1	**0%**							▨		
Document writting												
Paper	4	4	**0%**				▨	▨	▨	▨		
Thesis	7	2	**0%**							▨	▨	

Conclusion

Shuttle is the first:

- Intrusion recovery service for PaaS using replay
- To NoSQL databases and snapshoting
- To concern the parallel replay

Amongst the first:

- To incorporate the instance renewing
- To recover without application downtime

Thank you for your attention