NoSQL Undo: Recovering NoSQL Databases by Undoing Operations

David Matos, Miguel Correia
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisboa – Portugal
NCA 2016
PROBLEM AND MOTIVATION
Context

• NoSQL Databases are widely used in different applications

• NoSQL Databases are highly scalable

• NoSQL Databases are used in several Web Applications
Intrusions in NoSQL Databases

• Web Applications are vulnerable

• SQL Injection and XSS attacks create incorrect records in the database

• NoSQL Databases are not immune to attacks
Removing Intrusions in NoSQL DBs

• Removing intrusions is a complex and error prone task
• Restoring the last backup discards newer and valid information
• It is preferable to “correct” the state of an application instead of deleting the last block of operations
• Correcting the effects of intrusions is complex
• There are Intrusion Removal tools in the literature but they require software modifications
NOSQL UNDO
Rollback Recovery vs Undo

Database before recovery

After Rollback

After Undo
NoSQL Undo

- NoSQL Undo is a client-side application
- It connects to a NoSQL database
- Uses existent Database Logs to perform recovery
- It does not require any modifications to the database software
- It provides a Backup Service to extend the timespan of recovery
- It can be integrated with an IDS to facilitate the process of identifying malicious operations
NoSQL Undo Architecture

NoSQL Database Instance

Router

NoSQL Undo

Support Layer

Intrusion Detector

Global Log Backup Service

Shard 1
Primary
Secondary Secondary Secondary

Shard 2
Primary
Secondary Secondary Secondary

Config Servers
Config
Config
Recovery Algorithms

• Full Recovery
  – Simple algorithm that recovers every correct operation in the database

• Focused Recovery
  – More precise algorithm that only “corrects” the incorrect operations
Full Recovery

• Recovery in three steps (3 R’s):
  – Rewind
  – Repair
  – Replay

• If there are few incorrect operations to repair it becomes slow
Focused Recovery

• Recovers a database by applying correction operations to the affected records

• For each affected record:
  – Gathers every correct operation that affected that record
  – Calculates how the record would be if no incorrect operations took place
  – Substitutes the incorrect document with the correct one
IMPLEMENTATION WITH MONGODB
Implementation

• NoSQL Undo was implemented as a Java Application

• It was configured to recovery Mongo Databases

• Uses MongoDB Logs (used to maintain consistency)
EXPERIMENTAL EVALUATION
As the number of incorrect operations increases, Full Recovery becomes a better approach.
Recovering Multi Version Documents

Full Recovery maintains the TTR while Focused Recovery increases exponentially
The overhead of using an IDS is around between 10% to 30%
The overhead of the Backup Service varies from 6% to 30%
CONCLUSIONS
Conclusions

• NoSQL Undo is capable of removing the effects of undesirable operations in a database
• NoSQL Undo provides 2 methods to recover a database
• Focused Recovery is capable of removing an incorrect operation about 1 second
• If more than 60% of the operations in the log are incorrect, then Full Recovery is a better choice
Thank you

David Matos
david.r.matos@tecnico.ulisboa.pt

This work was supported by the European Commission through project H2020-653884 (SafeCloud) and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 (INESC-ID).