Confiabilidade de Sistemas Distribuídos
Dependable Distributed Systems

DI-FCT-UNL, Henrique Domingos, Nuno Preguiça

Lect. 6
Pro-Active Recovery Approach
and Intrusion Detection Systems

2015/2016, 2nd SEM

MIEI
Mestrado Integrado em Engenharia Informática

Last lecture (L5): Protection with IPSec and VPNs

- IPSec as a possible solution for orthogonal security: protection of channels at network level
- Protection in the layer-below the transport level channels (TCP or UDP based communication channels)
  - Transport mode vs. end-to-end security arguments
  - Tunneling mode
  - Different security properties (IPSec Stack, Sub-Protocols): AH, ESP-CA, ESP-C
  - VPN support
  - SAs / Composition of Sas
  - ISAKMP and IKE: establishment of SA parameters

Pro-Active Recovery vs. Intrusion Prevention vs. Intrusion Detection

Conjugation: Prevention > Detection > Recovery

- Intrusion Prevention Systems (IPS)
  - Preventive Solutions: including Firewall-based approaches, traffic shapers / blocking boxes, traffic inspection systems (with possible stateful inspection) in different typologies and configurations
    - We will discuss IPS and perimeter defenses **later**
- Intrusion Detection Systems (IDS)
  - Intrusion detection, relevant when IPS fail
  - IDS approach could allow a reactive intrusion recovery approach

Intrusion Recovery Approach

- Main idea in the context of intrusion recovery: How to remove intrusions, in such a way that:
  - The number of compromised servers (replicas) must be always below f
  - Maintaining availability conditions (avoiding a “stop the world” approach) and recovering compromised servers (replicas) in the life-cycle of system operation
  - How to discover compromised servers (replicas) ?
    - Use of Intrusion Detectors? Reactive Approach ?
      - Currently, this type of systems may not be able to be used for the objective (in order to preserve availability)
      - Some problems (remaining in the research agenda):
        - » Effectiveness, false positive / false negative rates (or base rate fallacy)
        - » Timing assumptions-constraints, “just in time” detection and recovery (circumvention of vulnerability window)
        - » Problem of Zero-Day vulnerabilities

IR: Intrusion Recovery

- Reactive Intrusion Recovery Approach
  - A reactive intrusion recovery solution can be fired by Intrusion Detectors (implemented by Software based solutions (Intrusion Detection Components) or orthogonal (vertical) dedicated IDS systems: HIDS, NIDS or Honeypots (or hybridized systems)
- SW based IR deals with different directions
  - From more generic to app-specific Intrusion Detectors

IR: Intrusion Recovery for Intrusion Tolerance

- Particularly interested in IR for Intrusion-Tolerant Distributed Systems (Dependable Systems)
  - Ex., based on SMR approach (leveraged by CONSENSUS protocols, with reliable and secure state-transfer support)
    - Possible use of deterministic or randomized consensus
    - Practical BFT protocols
  - IR Approach more related to **pro-active recovery**
  - Techniques can be conjugated with
    - Periodic Rejuvenation
    - Enhanced by Diversity
    - Randomization

Pro-Active Recovery for Intrusion Tolerance

- Idea: Periodically, a process for the rejuvenation of each server (replica) is fired, to achieve a correct state
- During the rejuvenation process:
  - All the malicious modifications which caused incorrect state or code tampering
  - But the rejuvenation is done, even when no intrusions take place

Problem …

… and rational on "independent" failures/intrusions (1)

- Problem with tolerating *f* faults:

  If an intelligent adversary is able to compromise *f* machines, given enough time, he/she will compromise *f+1* (or more)

  ⇒This is the base rational (starting point) for Proactive-Recovery [Castro&Liskov,TOCS2002]

  Replicas (compromised or not) are cleaned periodically, because soon or later they will be failed / attacked

Problem …

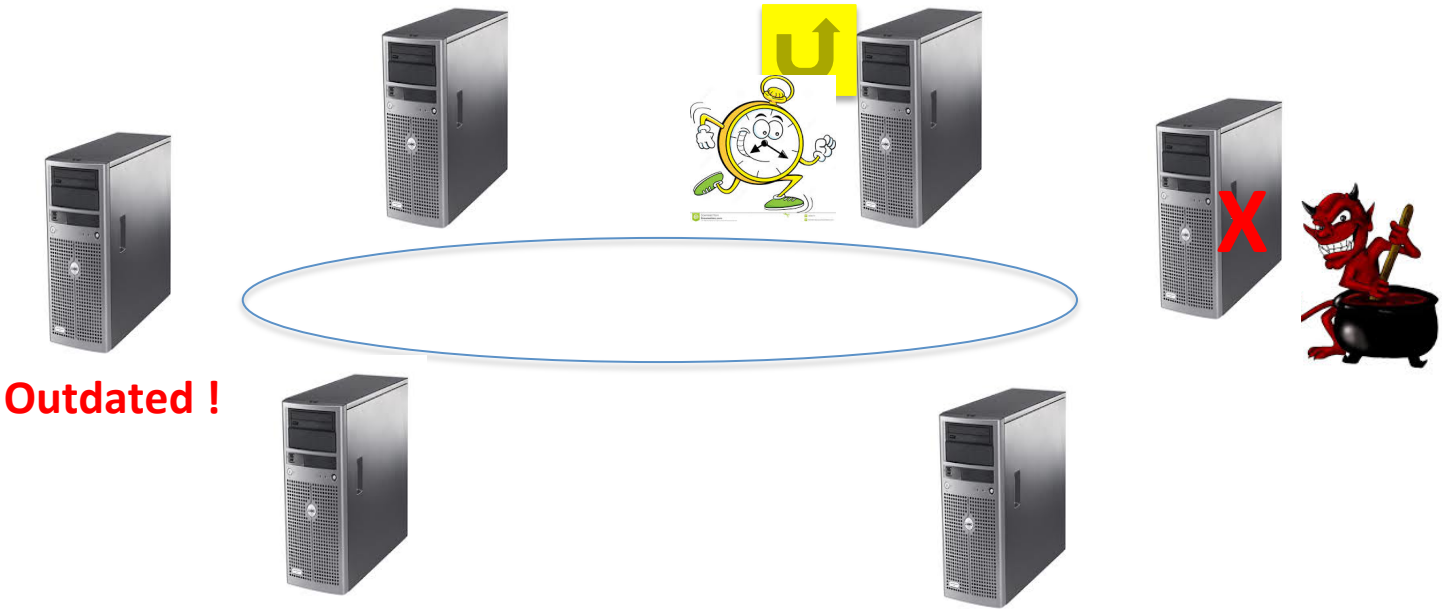
… and rational on "independent" failures/intrusions (2)

- But …
- PR requires local TCB components, anyway:
  - Trusted real-time component (ex., timer)
  - Possibly, trusted loader, Crypto module NVRAM and RO-Storage
    - TPM Assumptions: in HW TPMs
      - (ex., see TPM emergent 2.0 Assumptions)
- Otherwise: the PR process may be vulnerable to certain attack types
  - Currently, some pro-active recovery systems are vulnerable …

Problem …

… and rational on “independent” failures/intrusions (3)

- Other considerations
  - To ensure availability (“business continuity assumptions) you may also need *2k* extra replicas if at most *k* recover at the same time



Pro-Active Recovery for Intrusion Tolerance

- Important to notice:
  - The technique don't avoid the possible vulnerability of the system: the idea is to minimize the adversary hypothesis in compromising the security of the system
  - In practice: the risk of compromising more than *f* servers / replicas is circumvented to a vulnerability window that depends on the rejuvenation time
  - In Intrusion-Tolerance terminology, attackers trying to compromise servers, successively, is called a MOBILE ADVERSARY *

*) Ostrovsky, Yung, How to withstand mobile virus attacks, Proc. of 19th ACM Symposium on Principles of Distributed Computing, 1992

Complementary for BFT or BIT (Byzantine Intrusion Tolerance) enforcement: **DIVERSITY**

- *f*-fault-tolerant replicated systems are useful only if faults are not correlated
  - Independent Failure / Intrusion Model
  - No Collusion Attacks
- It usually requires **diverse replicas**

What is **DIVERSITY** about ?

- Different administrative domains
- N-version programming (effective?)
- Obfuscation, Memory randomization (effective?)
- Use of different components like databases (Gashi et al, TDSC 2007), file systems (Castro et al, TOCS 2003) and operating systems (Garcia et al, DSN'11) is effective!
  - Heterogeneous Software STACKS !
- *What about deploying and managing diversity?*
  - Good news: Virtualization, (Fast-Access) RO / Encrypted Flash Memory, SSDs, …

What more ? Randomization

- Randomization
  - Refreshing Replicas (and their Diverse Eco Systems) with Randomization Principles
  - Example: Randomized Chains of Diverse Firewalls !

Some relevant references …

- Abd-El-Malek et al. *Fault-scalable Byzantine Fault- tolerant Services*. SOSP'05
- Cowling et al. *HQ-Replication: a Hybrid Quorum Protocol for Byzantine Fault Tolerance*. OSDI'06
- Kotla et al. *Zyzzyva: Speculative Byzantine Fault Tolerance*. ACM TOCS 2009 (prel. SOSP'07)
- Guerraoui et al. The Next 700 BFT Protocols. EuroSys'10
- Amir et al. *Byzantine protocols Under Attack*. IEEE TDSC 2011
- Levin et al. TrInc: Small Trusted Hardware for Large Distributed Systems. NSDI'09
- Veronese et al. *Spin One's Wheels? Byzantine Fault Tolerance with a Spinning Primary*. SRDS'09
- Gashi et al. Fault tolerance via diversity for off-the- shelf products: a study with SQL database servers. IEEE TDSC 2007
- Garcia et al. OS Diversity for Intrusion tolerance: Myth or Reality? DSN'11

- M. Castro, B. Liskov, Practical Byzantine Fault Tolerance with Pro-Active Recovery, TOCS 2002
  - http://research.microsoft.com/en-us/um/people/mcastro/publications/p398-castro-bft-tocs.pdf
- Castro et al, **BASE: Using Abstraction to Improve Fault Tolerance**", *ACM Transactions on Computer Systems (TOC 2003)*
  - http://research.microsoft.com/en-us/um/people/mcastro/publications/p236-castro-base-tocs.pdf

PBFT-PR

- The PBFT Approach was extended to support a Pro-Active Recover approach
- The support includes 3 Base Operations:
  - Rekeying (renovation of secret keys used in the communication rounds C/S and S/S and in MAC computations/verifications)
  - Reposition of code (if compromised)
  - Reposition of correct state (if compromised)

*) M. Castro, B. Liskov, Practical Byzantine Fault Tolerance and Pro-active recovery", ACM Transactions on Computer Systems, 20(4):398-461, Nov 2002

PBFT-PR (solution)

- Requirements for each node:
  - A Cryptographic Coprocessor (storing the private key of the replica, and providing digital signatures and encryption/decryption without exposing keys
  - NV Read-Only memory, to store public keys of the other replicas, as well as, the recovery monitor (ex., BIOS)
  - A secure timer (trusted) to fire the recovery process (possible use of HW timers for this purpose)
  - Restrictions:
    - The adversary cannot have physical access to the node
    - Timing hypothesis: there is a certain instant t (unknown), after which the communication delay is below a given threshold value

Key-Refreshment

- In each period (ex, 1 minute) a new message with a new key is sent
- S_i sends to S_j:
  - {new-key, i, j, ….., $\{k_{i,j}\}_{KpubSj}$, ….., t $\}_{SIG,KPrivSi}$
  - $K_{j,i}$: used for HMACs sent from S_j to S_i
  - t is a sequence counter (protecting replaying)

  - HMAC Keys used for one-direction messages

  Communication with the client involves one key (bidirectional) and is distributed by the server (with a similar message as above

Code reposition

- This operation is fired by the trusted timer
- When the timer fires:
  - The recovery monitor creates a new code-image and teh state of the replica stored in disk
  - Forces a machine reboot
    - To verify if everything os OK it uses secure hash-proofs of SO and service code (SW atack to be reloaded) stored in read-only memory
    - If the SW stack is compromised, it must be necessry to obtain a copy of such images from other servers

State Reposition

- A protocol involving the other servers (replicas), to determine if the state is correct (or if it is compromised)
  - If compromised:
    - The new state is transferred from the other replicas
- Vulnerability window in the PBFT-PR

  Tv = 2Tk + Tr

  Tk: maxim period for rekeying

  Tr: recuperation period of the server

Another Pro-Active recovery approach: COCA System

- COCA means Cornell On Line Certification Authority
  - Motivation: Intrusion Tolerant CA (developed in the context of the OSASIS program)
    - Provide certificates with associations

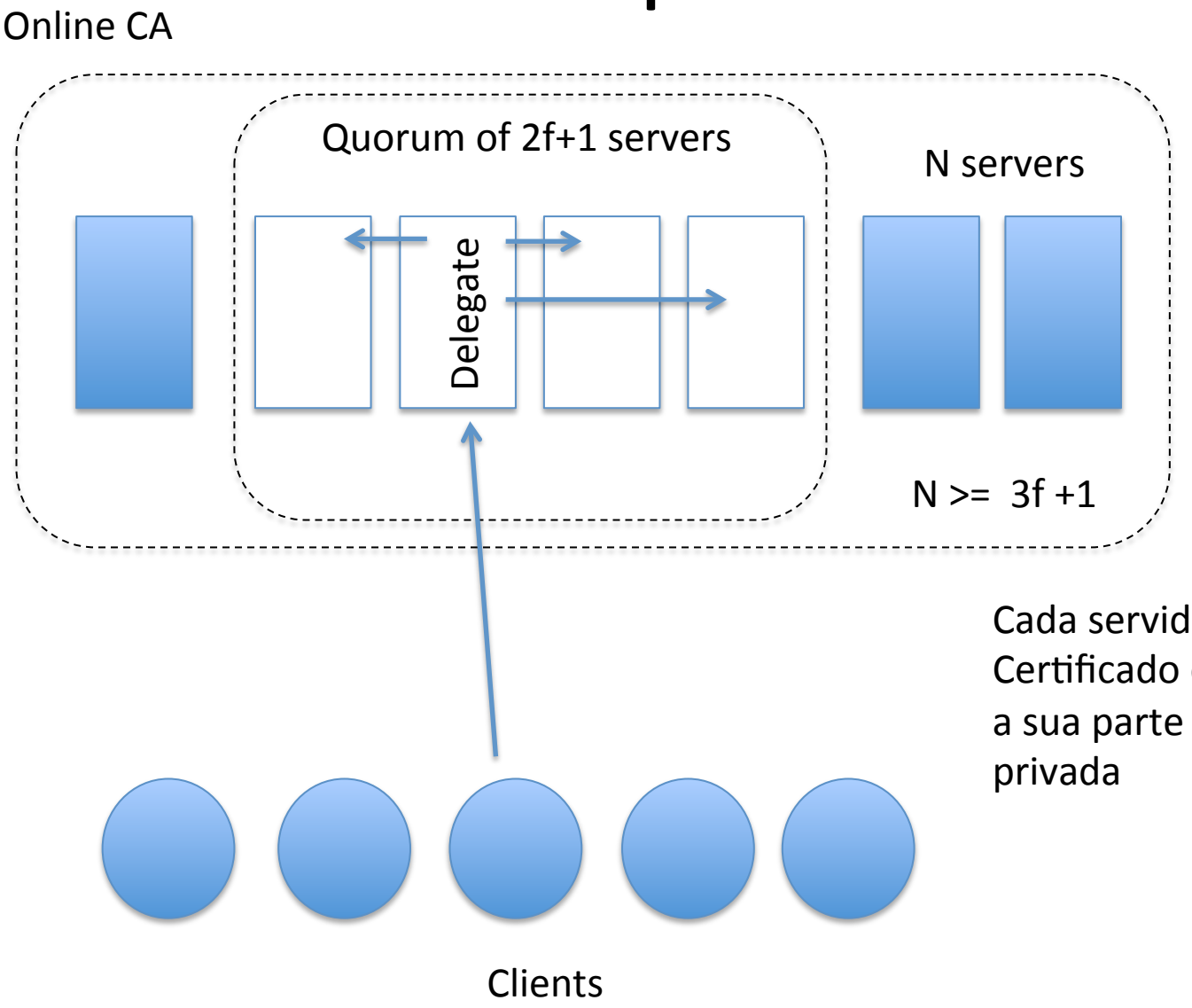
      <name, public-key>
    - Two base operations:
      - Update // to create, update or invalidate associations
      - Query // to obtain a certificate given a name
  - Approach: simplicity compared with PBFT-PR
  - Use of dissemination quorums, N >= 3f+1, Quorum Size = 2f +1

COCA System

- Use of (k,N) threshold-signature cryptographic construction (asymmetric crypto scheme), with k=f+1
- All clients and servers know the public key of the service, but the private key is distributed by all the servers (as private key shares)
- A certificate signature requires a quorum k

See:
https://en.wikipedia.org/wiki/Threshold_cryptosystem

COCA Operation



Cada servidor obtem o Certificado e assina com a sua parte da chave privada

COCA Pro-Active Recovery

- 3 operations:
  - Refreshment of the Private Key Shares for each server
  - Code reposition (if compromised)
  - State reposition (if compromised)

Similar to the PBFT-PR approach

The idea here is to avoid a MOBIEL ADVERSARY to compromise f+1 servers
In order to capture the f+1 private key shares

Refreshment based in a pro-active protocol working as a
Proactive protocol for secret sharing (COCA uses the APSS Algorithm)

COCA Pro-Active Recovery and other Related Papers

COCA and APSS

L. Zhou, F. Schneider, R. Van Renesse, COCA: A Secure Distributed On-Line Certification Authority, ACM Transactions on Computer Systems, 20(4): 329-368, Nov 2002 (suggested reading)

More …

- L. Zhou, F. Schneider, R. Van Renesse, "Pro-Active Secret Sharing in Asynchronous Systems", TR 1877, Cornell University, Oct 2002
- C. Cachin, K. Kursawe, A. Lysyanskaya, R. Strobl, Asynchronous Verifiable Secret Sharing and Pro-Active Cryptosystems, Proc. 9th ACM Conference on Computer Communications Security, 2002
- M. A. Marsh and F. B. Schneider, CODEX: A Robust and Secure Secret Distribution System, IEEE Transactions on Dependable and Secure Computing, 1(1): 34-47, Jan-Mar, 2004

More on Pro-Active Recovery

- Approach to a solution for TP2
  - More (later) in the discussion of TP2 requirements and objectives, possibly will involve the design, implementation and evaluation a pro-active recovery mechanism, as a new work direction in evolving the initial TP1 implementation
    - Together with other requirements that will be added