Intrusion Tolerant SCADA

Nuno Neves

(join work with André Nogueira and Alysson Bessani)

LaSIGE, University of Lisboa, Portugal

(www.navigators.di.fc.ul.pt)

Symposium on Innovative Smart Grid Cybersecurity Solutions
Vienna, March 2017
SCADA: Example Industrial Facilities
Master SCADA Server Tolerant to Crashes

- Redundancy method with two identical replicas to tolerate crashes
- Upon failure of the primary server, the backup replica takes over, replacing it
- The backup replica can operate as a hot/warm/cold standby
- The backup replica needs to be kept updated

- Tolerates one crash fault
- Relatively short downtime period
Scenarios of SCADA under Attack

- Reported advanced cyber attacks continue to rise
  - this is particularly visible in the electrical area with many described incidents

- Add intrusion prevention using standard IT security technologies
  - Firewalls
  - IDS

Given the criticality of the SCADA system it is **prudent** to assume that prevention is not perfect!
How to make a system intrusion tolerant?

Main ingredients:
• High level of mistrust in all components
• Assume a bound on the number of replicas compromises

Use State Machine Replication (or Active Replication) to tolerate arbitrary/Byzantine faulty behavior
• Weakest possible failure assumption
  \[ n = 3f + 1 \quad (f=1, \; n=4) \]

1. Every client request is processed by a group of servers
2. Servers must execute the same sequence of requests
3. The client infers the correct result of a request from the majority of the answers

• Servers coordinate to order the request processing
• Servers should run diverse software
Putting the Idea into Practice!

• Develop an intrusion tolerant SCADA Master server
• Integrate an active replication library with an open source SCADA
• Achieve a performance level to be used in the field

Some of the challenges:
• Find a suitable open source software → EclipseSCADA
• Project size (more than 500 sub-projects → 900,000 LOC)
• Limited documentation
  ➢ source code
  ➢ use case examples
• EclipseSCADA is a framework and not a ready-to-use solution
EclipseSCADA: Overview of Components

**Frontend**
- **DA Server**
- **Item**

**SCADA Master**
- **DA Client**
- **Item**
- **Handlers**
- **Storage**
- **AE Server**

**HMI**
- **DA Client**
- **Item**

**Performs Data Acquisition (DA) operations such as SubscribeItem, WriteValue,…**

**Maps the frontend’s items**

**Performs DA operations such as SubscribeEventItem, SubscribeMonitorItem,…**

**Gets Alarms & Events (AE) that were setup**

**Records events related with items**

**Provides additional functionalities to items(Scale, Block, Monitor,…)**

**Maps the master’s items**

**Represents a single value provided by a device (it may contain attributes)**
Challenges for replication

- Multiple I/O channels
- Concurrency of processing
- Timestamps
- Asynchrony of message arrival
- Performance
- Minimize changes in the code
Resilient EclipseSCADA

Channel to place EclipseSCADA messages that go to/come from the Frontend

The SCADA master server needs to be replicated

Demultiplexer & forwarding & ordering EclipseSCADA messages
Preliminary Evaluation

• Measure the **impact in performance** introduced by active replication
  ➢ compare the performance of the **EclipseSCADA** and the **BFT EclipseSCADA**

• Hardware setup infrastructure
  ➢ 1 Frontend
  ➢ 4 EclipseSCADA Master Servers
  ➢ 1 HMI
Item update test

Benchmark:
Update 10 items in the frontend
Each item updates itself 100 times per sec

Overhead of 6% mainly explained due
• 2 commun. steps in EclipseSCADA, 6 steps in BFT EclipseSCADA
Item updates with alarms test

**Benchmark**
Update 10 items in the frontend
Each item updates itself 100 times per sec
Add monitor handlers

- Overhead of 25% in worst case scenario (with 100% alarms)
Item writing test

Benchmark:
Write synchronously 1 item in the frontend

Overhead of 78%
- **4 steps** in EclipseSCADA, **12 steps** in BFT EclipseSCADA
- The **active replication library** introduces an overhead of only **12%**
Thank you! Any questions?

This was:

Intrusion Tolerant SCADA

Nuno Neves  (nuno@di.fc.ul.pt)
Web: www.di.fc.ul.pt/~nuno  and  
www.navigators.di.fc.ul.pt