

Rome, 28 February, 2011

# Usage of trust and policies in the exchange of information between Critical Infrastructures

F. Caldeira  
FCTUC



## Outline

- Motivation
- SMGW Policy Based Manager
- Trust indicators in the MICIE project
- Results
- Conclusions

## Motivation

- ICT security plays a major role in CI protection and risk prevention for single or interconnected CIs, where cascading effects might occur due to the existing interdependencies.
  - In a scenario where multiple CIs are willing to exchange risk information and to incorporate risks from interdependent services in its own risk level, we propose to:
    - Use a Policy Based Management Architecture.
    - Use this **Policy Based Management**, along with **Trust and Reputation Indicators** at the CI interconnection points for information exchange, to improve the system behaviour.



## Motivation

- How do we manage connections and information sharing authorizations?
- Can we trust on the information received from peer CIs?
  - Monitoring components can be faulty
  - The peer CI ICT system can be compromised, sending false information
  - The peer CI may intentionally provide inaccurate information
- Is the behavior of peer CIs acceptable?
  - Has the monitoring framework been compromised?
  - Repeatedly trying to read non-authorized information?
- Answers to this questions are important as
  - The CI uses received alerts to infer its own risk levels
  - Trusting false information affects risk assessment
  - Shared information is highly confidential

## SMGW Manager

- CI operators can define, in a high level manner, the intended system behaviour.
- Handles AAA for both internal and external operations.
- Policies:
  - address the relations between the local SMGW and foreign SMGWs.
  - are represented in a formal way and stored in policy repositories.
- The SMGW manager:
  - interacts with other entities on the SMGW using a Web Service API.
  - Maintains a real time image of all system.
  - Implementation based on the Ponder2 toolkit

## Policy SMGW Manager - Key Components:

- Policy Database (XML) – stores all defined policies
- Policy GUI – interface with the administrator of the SMGW
- Migration Tool
  - Allows the migration of data in the SMGW Database to the policy manager (remote CI names, risk data names, basically all existent tables/attributes can be mapped to the manager).
- SMGW Manager/Policy-based Management:
  - The SMGW Manager acts as the system *Policy Decision Point* (PDP)
  - It is the SMGW component responsible for authorizing access to the SMGW and to its data. Decisions are based on stored policies and context information.
- PEP - *Policy Enforcement Points*:
  - PEPs control the access to the SMGW (communications, data) based on the rules received from the SMGW Manager. These may include, for instance, firewalls, VPN servers, Web Servers and Data Access Services.
- Reputation Service – an optional module to monitor and estimate the reputation and trust level of partner CIs (*to be discussed later*)



## Policy Manager - Implementation

- Server / Manager GUI

The screenshot displays the MICIE Policy Manager GUI. On the left, a vertical panel contains a 'Server Started' status message and an 'Alerts come here' section. The main window has a tabbed interface with the following tabs: 'Add Auth Policy', 'Manage Auth Policies', 'Add Event Policy' (selected), 'Manage Event Policy', 'New Managed Object', 'Manage Objects', 'Test', and 'Domains'. The 'Add Event Policy' tab contains the following fields and controls:

- Policy Path:** A text input field containing 'value'.
- Event Attributes:** A text input field containing 'value'.
- Event Condition:** A text input field containing '[:risk | risk>5]'. To its right is a circular refresh icon with blue and green arrows.
- Event Action:** A text input field containing '[ root/alertEngine alert: "ExampleAlert"]'.
- Policy Activated:** A dropdown menu currently set to 'true'.
- Create Policy:** A button located at the bottom left of the main form.

On the right side of the 'Add Event Policy' tab, there is a tree view showing the hierarchy of managed objects:

- Root: /
- Child: /domain1
  - Child: /domain1/object1 (represented by a green square)
- Child: /domain2
  - Child: /domain2/object2 (represented by a green square)
- Child: pol1 (represented by a black square)
- Child: p1 (represented by a red square)



## Policy Manager - Implementation

- Clients(for testing Purposes)

MICIE - Critical Infrastructure Client - Demo

File

Basic Advanced Log

Request Managed Object

Source Domain:

Source Object:

Target Domain:

Target Object:

Managed Object Attributes

name	value

Settings

Site:

Port:

AppPath:

Resource:

MICIE

File

Source

Target





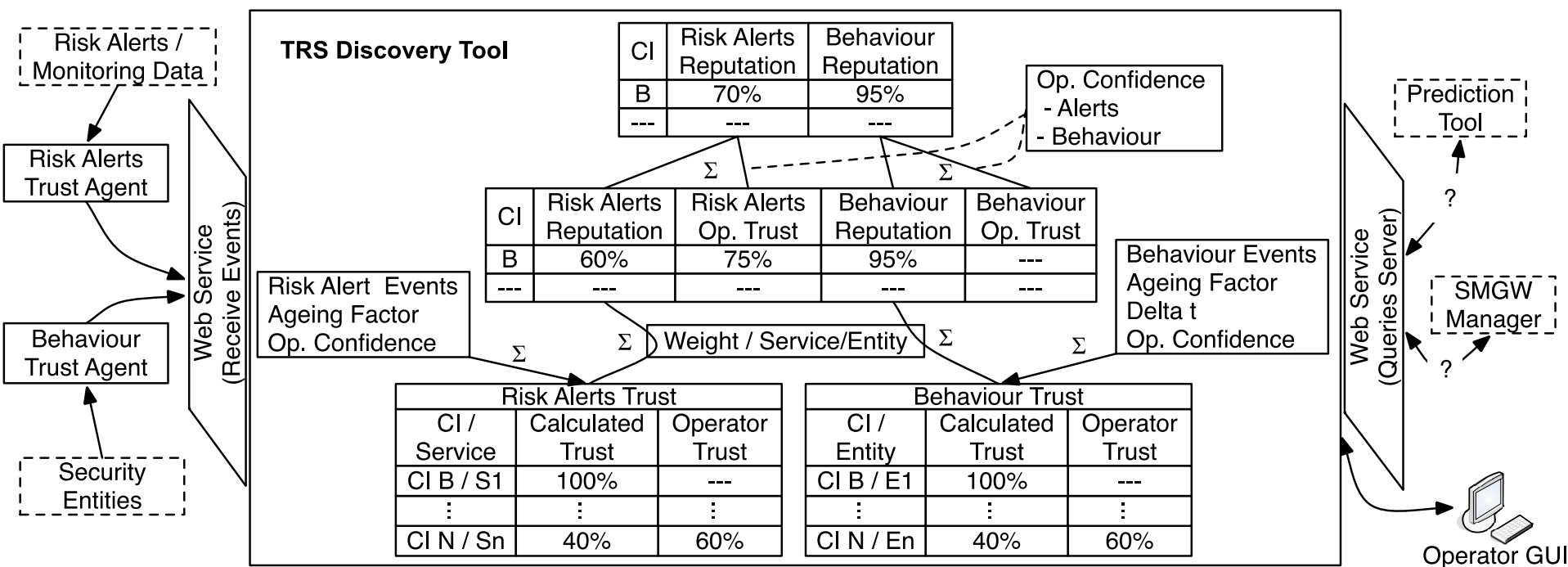
## Trust and Reputation Service (TRS)

- An extension to the MICIE core framework.
- The Trust and Reputation Service allows to:
  - associate a level of trust to the data received from peer CIs, as well as to its own internal monitoring data (e.g. SCADA systems).
  - use trust levels to enhance the accuracy of the MICIE Risk Prediction Tools.
  - detect defective components (at local level) which consistently provide inaccurate information.
  - detect partner CIs that systematically provide inconsistent information.
  - Incorporate trust indicators in:
    - The risk assessment tools (limiting the impact of inconsistent information)
    - The access policies of the MICIE framework (limiting the access of non-trusted partners to sensitive data)

## **Reputation Service – input sources**

- Analysis of past data provided by partner/“service”
  - Each partner CI provides and/or receives a number of “services” (the interdependency links)
  - For each provided “service” the partner CI also provides a risk assessment estimate, related to its availability or QoS.
  - Compare the risk estimates provided over time, for each “service”, against the actual service levels provided over time, in order to infer the trustiness of future estimates.
- Analysis of partner behavior
  - E.g. if the partner CI behaves abnormally (for instance trying to access non-authorized data or using non-authorized credentials) downgrade the global level of trustiness associated with that partner CI.
- Human factor (Optional)
  - Incorporate the perception of the human operator about the trustiness of each peer CI or each “service”.

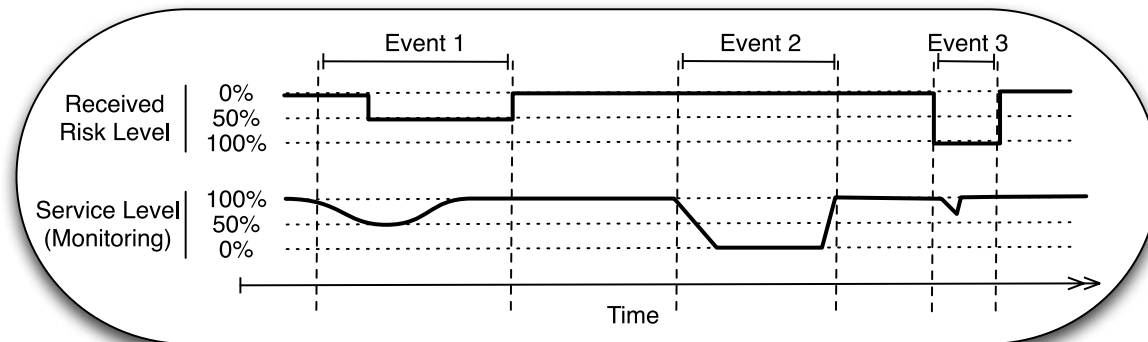
## Trust and Reputation Service



## Trustiness of Received Alerts (1)

An Event is triggered:

- (1) When one service decreases its Quality of Service (QoS) below its predefined threshold.
  - the event ends when the QoS reaches the threshold level again
  - if an alert is received, the event ends when the alert is removed;
- (2) during the period of a risk alert message.





## Trustiness of Received Alerts (2)

- The Event Accuracy is defined as the average of all comparisons made during the event.
- The trust that one CI has in the risk estimates received for a “service” provided by another CI is based on the accuracy of each past event involving those two CIs (for that specific “service”), with two factors that depend on the context:
  - Penalization factor (penalize larger estimation errors)
  - Aging factor (weight recent events more than old events, applying an aging factor to each event).

## Trust on Received Alerts (3)

- Event accuracy:

$$A(Event_n) = \frac{\sum_{t=1}^T (f(Sl_t, Rl_t))}{T}$$

$$f(Sl_t, Rl_t) = |Sl_t - Rl_t|^\kappa, \kappa \in R^+$$

- Trust that CI has for service X provided by CI B :

$$T'_{(A,B,X)} = \frac{(D * (N - 1) * T_{(A,B,X)}) + A(Event_N)}{D * (N - 1) + 1}$$

$$T(final)_{(A,B,X)} = \alpha(T_{(A,B,X)}) + \beta(TO_{(A,B,X)}) , (0 < \alpha, \beta < 1), (\alpha + \beta = 1)$$

- K - Penalization factor (penalize more bigger differences)
- D - Aging factor (weigh recent events more that old events using a discount factor applied to each event).
- N – Event Number
- T(final) – Incorporating Operator trust (TO)

## Global Trust - Multiple Services

- With the help of the expected results from MICIE project on risk modelling and analysis, we can weight each service according to that analysis,
  - for instance, giving more weight on more critical services and less weight to services that have less impact on our CI.

$$GT'_{(A,B,t)} = \frac{(D * (N - 1) * GT_{(A,B)}) + \frac{\sum_{i=1}^S (T(final)_{(A,B,i)} * W_i)}{\sum_{i=1}^S W_i}}{D * (N - 1) + 1}$$

$$GT(final)_{(A,B,t)} = \theta(TO_{A,B}) + (1 - \theta)(T_{(A,B)}) \quad , (0 < \theta < 1)$$

- D - Aging factor (weigh recent events more that old events using a discount factor applied to each event).
- N – Event Number
- S – Number of services
- Wi – Service i weight
- GT(final) – Incorporating Operator trust (TO)

## Trustiness of Peers Behavior

- MICIE already foresees the collection and analysis of data related to security aspects of the SMGW
  - It is possible to use this valuable information in order to infer a Trust indicator for the behavior of each peer CI.
- Normalization of received information - Security model

Failed Authentication Attempts/Minute		
Trust Indicator Level	Description	Received Values
100	No Failures	0
80	One/Three Failures	1-3
20	Four/Ten Failures	$> 3$ and $< 10$
0	More that 10 Failures	$\geq 10$

- Security Indicators can be evaluated based on:
  - Intrusion Detection System
  - QoS measurements
  - Monitoring Systems



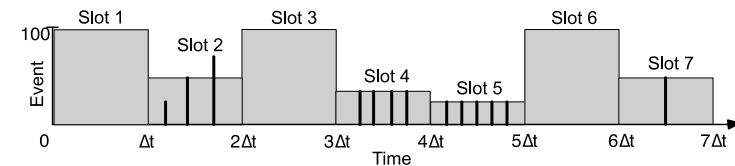
## **Trustiness of Peers Behavior**

- In order to consider aging in trust calculation, we consider time as a set of time slots. Each time slot will have a defined duration and represents an Event, in the calculations.
  - No activity in one time slot means that peer behavior was as expected, so the maximum value should be given to this event.
  - If alarms are received during the time slot, the estimated value for the slot will take into account all events that took place.
  - Size of time slot depends on the context.
  - An aging factor is also applied.
  - A Penalization factor may also be used.

## Trustiness of Peers Behavior

- In order to consider time in trust calculation, we consider time as a set of time slots, each one representing an event.
  - Size of time slot should be dependent on the context

$$Event_{(Slot\ s)} = \begin{cases} 100, & \text{if } NEvents_{(Slot\ s)} = 0 \\ \frac{\sum_{i=1}^N Event_i}{N}, & \text{if } N = NEvents_{(Slot\ s)} > 0 \end{cases}$$



- Trust on peers behavior:

$$T'_{(E,B,s)} = \frac{(D * (s - 1) * T_{(E,B)}) + Event_{(Slot\ s)}}{D * (s - 1) + 1}$$

$$T(Final)_{(E,B)} = \theta(TO_{(E,B)}) + (1 - \theta)(T_{(E,B)}) \quad , (0 < \theta < 1)$$

- D - Aging factor (weigh recent events more that old events using a discount factor applied to each event).
- t – Time
- T(final) – Incorporating Operator trust (TO)

## Trust on Peers Behavior

- Global Trustiness of CI Behavior:

$$TBehaviour'_{(B,t)} = \frac{(D * (t - 1) * TBehaviour_{(B)}) + \frac{\sum_{i=1}^E (T(Final)(i) * W_i)}{\sum_{i=1}^E W_i}}{D * (t - 1) + 1}$$

- D - Aging factor (weigh recent events more that old events using a discount factor applied to each event).
- t – Time
- E – Number of security entities
- Wi – Entity i weight
- $T_{Behaviour}(final)$  – Incorporating Operator trust (TO)



## Trust & Reputation Service - Implementation

- Main Application

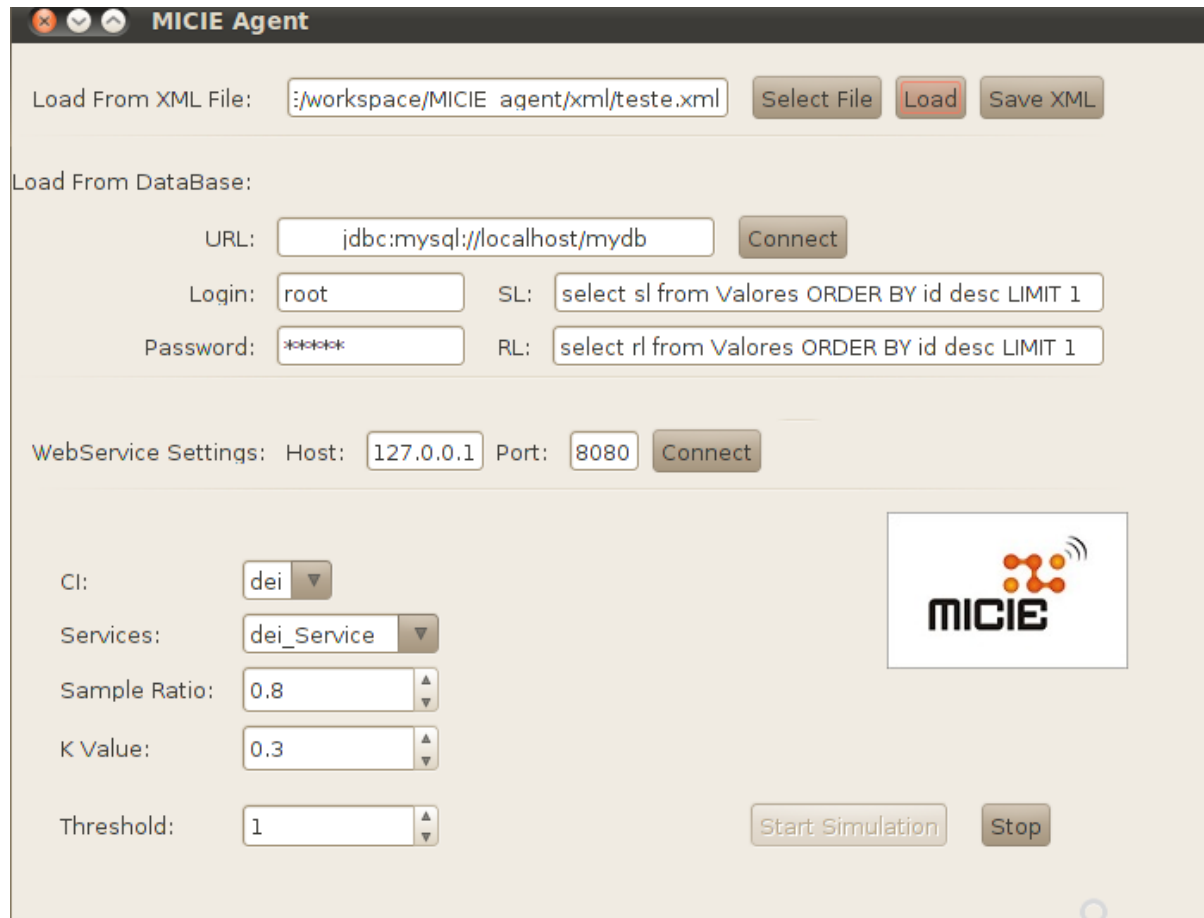
A screenshot of the MICIE Project application window. The window has a title bar with standard OS controls and the text "MICIE Project". Below the title bar is a menu bar with options: File, Charts, Insert, Reset, Remove, and Help. The main content area features a tabbed interface with tabs for Alert Trust, Reputation, Behaviour Trust, CI Settings, Service Settings, and Entity Settings. The Reputation tab is active, displaying a table with the following data:

CI Name	Service Name	Calculated Trust	Operator Trust	Final Trust
Dei	S1	0.798	0	0.798

Below the table is a large empty rectangular area. At the bottom left of the window is a smaller version of the MICIE logo. To its right, the text "MICIE Reputation Service" is displayed. At the very bottom left, the status "Refresh Finished" is shown next to a circular progress indicator.

## Trust & Reputation Service - Implementation

- Agent



The screenshot shows the MICIE Agent web interface. It features a title bar with the text "MICIE Agent". Below the title bar, there are several sections for configuration:

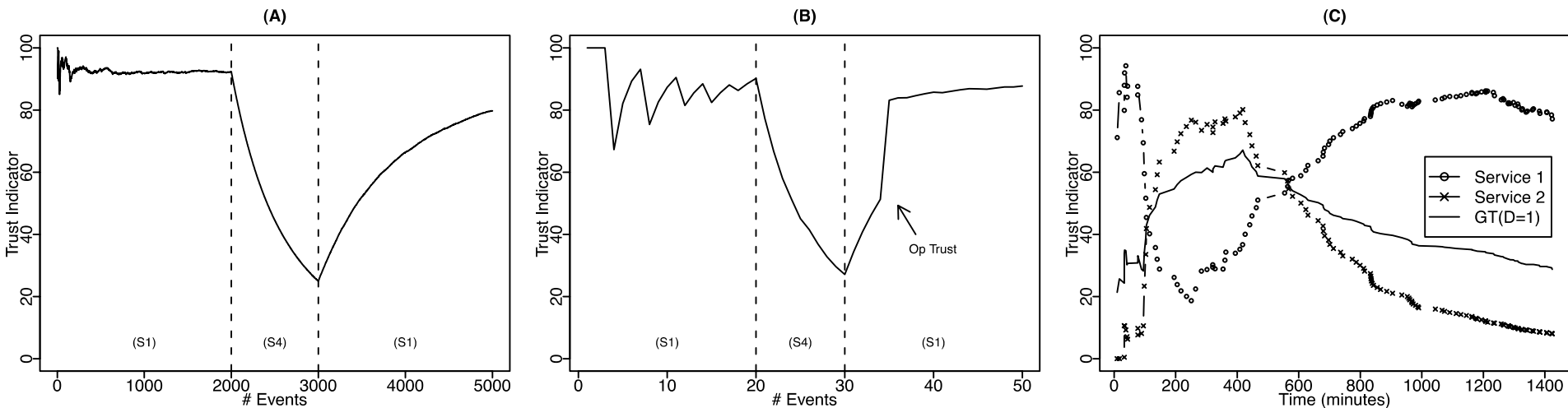
- Load From XML File:** A text input field containing the path `:/workspace/MICIE agent/xml/teste.xml`, followed by buttons for "Select File", "Load", and "Save XML".
- Load From DataBase:** A section with fields for "URL:" (containing `jdbc:mysql://localhost/mydb`), "Login:" (containing `root`), "Password:" (containing masked characters), "SL:" (containing `select sl from Valores ORDER BY id desc LIMIT 1`), and "RL:" (containing `select rl from Valores ORDER BY id desc LIMIT 1`). A "Connect" button is located to the right of the URL field.
- WebService Settings:** A section with fields for "Host:" (containing `127.0.0.1`) and "Port:" (containing `8080`), followed by a "Connect" button.
- Configuration Parameters:** A section with several dropdown menus and input fields: "CI:" (containing `dei`), "Services:" (containing `dei_Service`), "Sample Ratio:" (containing `0.8`), "K Value:" (containing `0.3`), and "Threshold:" (containing `1`).

At the bottom right, there are two buttons: "Start Simulation" and "Stop". A small MICIE logo is also visible in the bottom right corner of the interface.

## Validation

- Simulation tests:
  - Events are generated using a normal distribution
  - Threshold = 10% (trust values above 90% are rated as 100%)
  - Consider 4 scenarios:
    - (S1) The system behaves as expected with only small errors, with the event accuracy always above 60% and mainly between 90% and 100%.
    - (S2) System is not accurate but can still be trustworthy, as evaluated event accuracy is always above 40%.
    - (S3) Received alerts are not as expected with above 40% of inaccurate indications but never rising above 60%.
    - (S4) System is inaccurate.

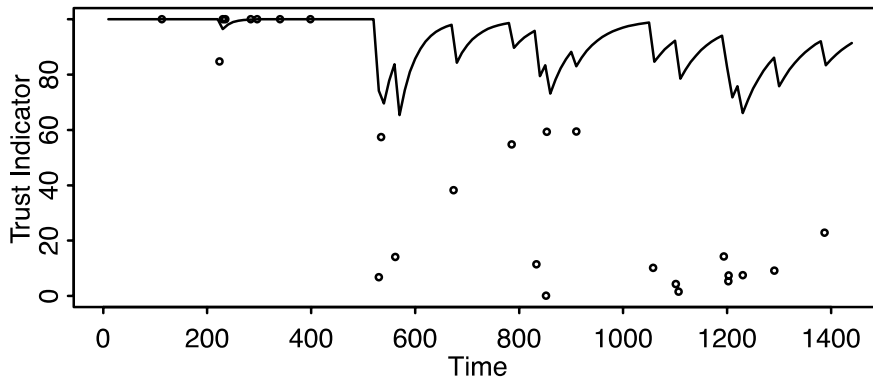
## Validation: attack or faulty component situation



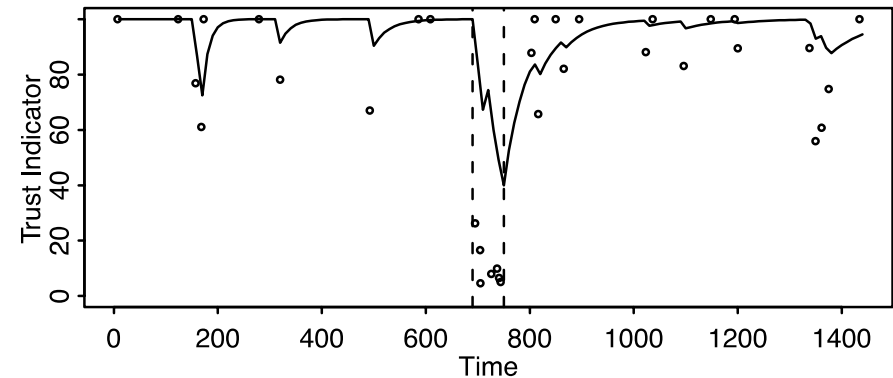
- Figures (A),(B) - it is noticeable that the trust indicator decreases rapidly and next starts to gradually grow depending on the scenario.
  - Figure (B) describes the use of the Human Factor showing that the operator can rapidly change the trust in a service.
- Figure (C) - Information observed from two services (weights: Service 1: 0.3 / Service 2: 0.7 )
  - In this simulation, when the service more important is becoming unreliable, then the CI reputation is decaying even when the other service is trustworthy.

## Validation: Trust on Peers Behaviour (1)

(C)



(D)



- Fig (C) has a rate of 1 event per hour from scenarios mixed scenarios.
  - With few events, the indicator does not drop below 60% - influence of the slots where the system is behaving well. Demonstrate how important the values defined for the time slot are.
- Fig (D) - Possible attack or misbehaviour in a small period of time.
  - The first 11.5 hours, exist 1 event/hour from one normal scenario; During one hour, exists 5 event/hour from a scenario where the peer behaviour is not normal.
    - The trust indicator rapidly decays below 50% clearly indicating that something is wrong.
  - Last simulated hours represent a normal scenario with an event rate of 1/hour.
    - The trust indicator clearly indicates the resolution of the past situation.





## Conclusions

- The proposed framework helps to answer questions like *“how much can we trust in received risk alerts or in the peer CIs behaviour?”*.
- Trust and Reputation indicators can be incorporated in CI risk assessment as a means to improve its accuracy and its resilience to inconsistent information provided by peer Critical Infrastructures.
  - It is possible, for instance, to give more weight to highly trusted data or to ignore data provided by non-trustable partner
- System managers can act more dynamically using trust and reputation indicators and reacting autonomously when those indicators change.
  - For instance, if our trust on one peer decreases below a defined threshold a new policy is triggered and the SMGW stops accepting connections from that peer.
- The Policy Based SMGW Manager is integrated in the MICIE SMGW.
- TRS is developed and has already been validated using simulation.

## Questions and Comments

