

# **POSITIF Workshop**

**October 16, 2006  
at Secure 2006 Conference (Warsaw)**

## **SDL & SPL LANGUAGES AND CONSOLES**

**Marcin Wojtkiewicz**

**[Marcin.Wojtkiewicz@pwr.wroc.pl](mailto:Marcin.Wojtkiewicz@pwr.wroc.pl)**

**Wrocław Centre for Networking and Supercomputing  
Wrocław University of Technology**

# Agenda

- SDL language overview
- Examples of the network topology description in SDL
- SPL language overview
- Examples of policies description in SPL
- SDL and SPL consoles

# SDL Overview

# SDL features

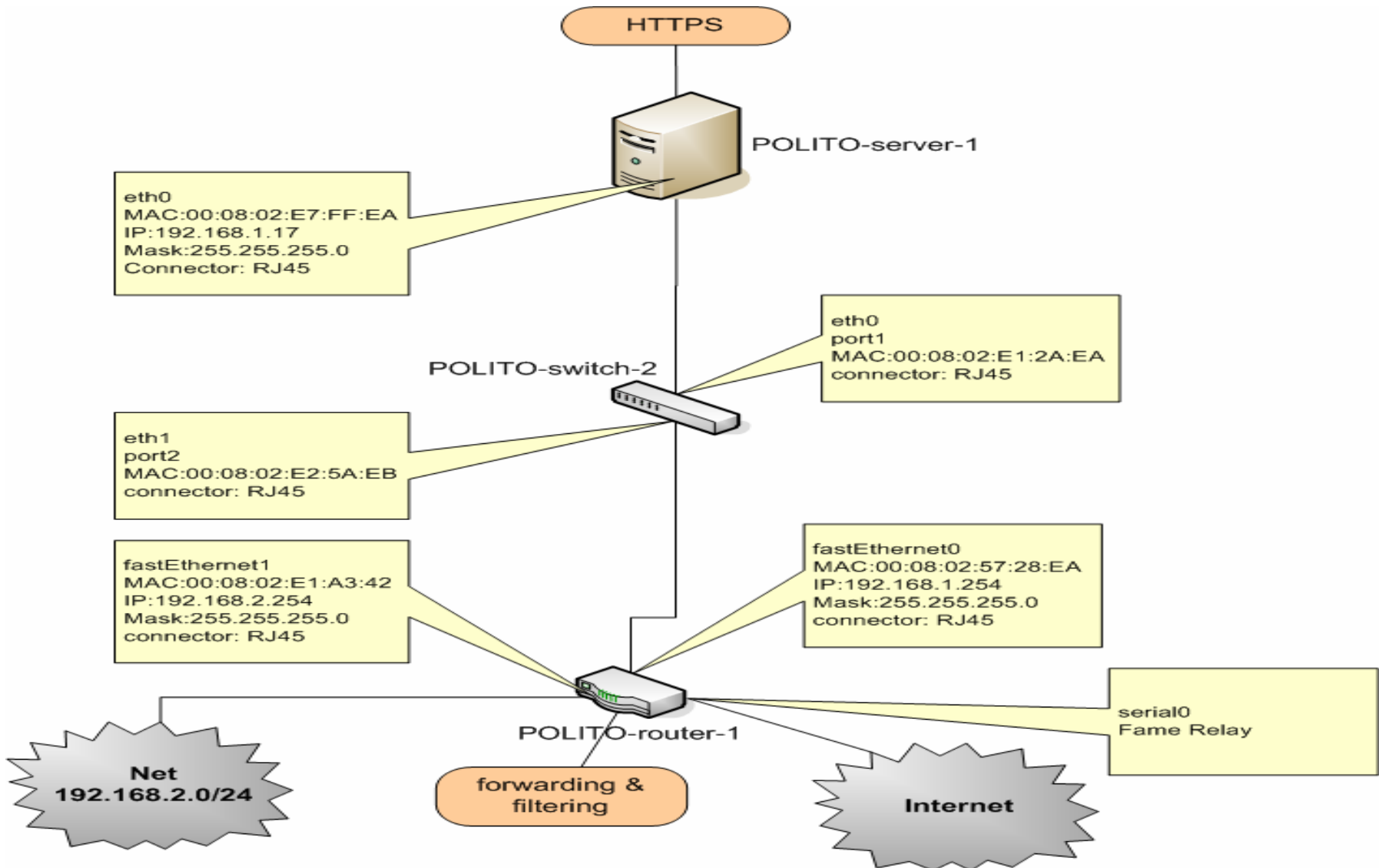
- Enables description of the topology of the system
- Describes the black-box functionality of each element (i.e. the network services offered and the applications supported)
- Describes the security functionality of the element (i.e. its available internal security features, such as network filters, OS intrinsic controls)

# Network topology in SDL

- The network will be represented by an XML document with the root element network
- The network elements and the links between them are child elements of the network element, as well as the link to other networks

```
<network id="polito" ...>  
  <router> ... </router>  
  <switch> ... </switch>  
  <connection> ... </connection>  
  <computer> ... </computer>  
  <operatingSystem> ... </operatingSystem>  
  <software> ... </software>  
  <networkAccessPoint> ... </networkAccessPoint>  
</network>
```

# Complete example of infrastructure in SDL



# Complete example of router in SDL

```
<router id="R1">
  <interface id="eth0" number="1" technology="Ethernet"
connector="RJ45" protocol="10-100BaseT">
    <addr type="hw">00:08:02:57:28:EA</addr>
    <addr type="ipv4" netmask="255.255.255.0">192.168.1.254</addr>
  </interface>
  <interface id="eth1" number="2" technology="Ethernet"
connector="RJ45" protocol="10-100BaseT">
    <addr type="hw">00:08:02:E1:A3:42</addr>
    <addr type="ipv4" netmask="255.255.255.0">192.168.3.254</addr>
  </interface>
  <interface id="eth2" technology="Frame Relay" connector="RS232">
    <addr type="ipv4" netmask="255.255.255.0">192.168.2.254</addr>
  </interface>
</router>
```

# Example of switch in SDL and its connection to the router

```
<switch id="S1" ifaces="16">
  <interface id="port01"
    number="1" technology="Ethernet" protocol="10-100BaseT">
    <addr type="hw">00:08:02:E2:5A:EB</addr>
  </interface>
  <interface id="port02"
    number="2" technology="Ethernet" protocol="10-100BaseT">
    <!-- unknown MAC address -->
  </interface>
</switch>

<connection>
  <endpoint elementId="S1" interfaceId="port01" />
  <endpoint elementId="R1" interfaceId="eth0" />
</connection>
```

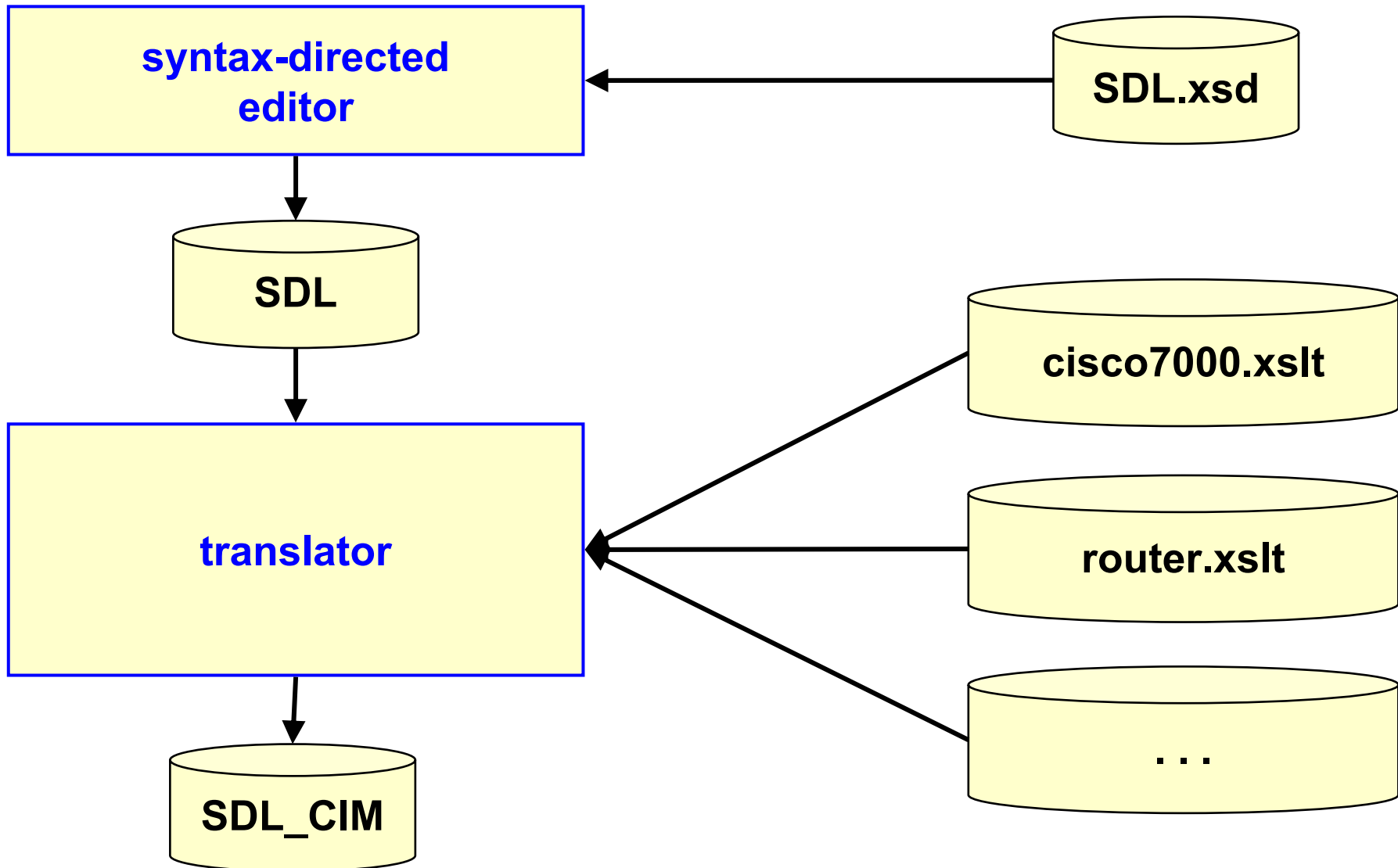
# Example of server providing http and ftp services in SDL

```
<computer id="server1">
  <interface id="eth0" technology="Ethernet">
    <addr type="hw">00:08:02:E7:FF:EA</addr>
    <addr type="ipv4" netmask="255.255.255.0">
      192.168.1.17</addr>
  </interface>
  <os idRef="MyOS" />
  <service id="web server">
    <protocol idRef="myhttp"/>
    <sap addr="192.168.1.17" transport="tcp" port="80"/>
    <software idRef="MyApache"/>
  </service>
  <service id="ftp server">
    <protocol idRef="myftp"/>
    <sap addr="*" transport="tcp" port="21"/>
    <software idRef="Myftpd"/>
  </service>
</computer>
```

# Example of server providing http and ftp services in SDL (cont.)

```
<software id="MyApache">
  <product>Apache </product>
  <version>2.0.1</version>
</software>
<software id="Myftpd">
  <product>Vsftpd </product>
  <version>1.3</version>
</software>
<protocol id="myhttp" name="http">
  <version>1.0</version>
</protocol>
<protocol id="myftp" name="ftp">
  <version>1.0</version>
</protocol>
<operatingSystem id="MyOS">
  <product>solaris</product>
</operatingSystem>
```

# Further SDL transformations



# SPL Overview

# SPL features

- SPL is based on CIM and derived from xCIM schema
- Enables definition of filtering, authentication, authorization, confidentiality and operational policies.
- Composed of an XML schema for each type of security policy.
- CIM-SPL instances are valid instances for the internal format.
- Links to SDL elements described by internal format.

# SPL examples

- Filtering policy: For the firewall of the Computer Science Faculty, allowing only outbound traffic for HTTP
- Authentication policy: For the file server of the Computer Science Faculty, users must authenticate with the shared secret named as UserFileServer
- Channel protection policy: For the file server of the Computer Science Faculty, the http connections must be SSL connections

# SPL examples

- Authorization policy: The students of Computer Science are allowed to print in the print server of the Computer Science Faculty.
- Operational policy: If the firewall of the Computer Science Faculty is failing or has failed, notify the Administrator via SMS.

# SPL examples

## ■ Examples:

- Example of authorization policy: The students of Computer Science are allowed to print in the print server of the Computer Science Faculty.

## ■ Elements:

- System elements: students (identities) and print server (system)
- Policy elements: role and privilege

# SPL examples

- System elements:

```
<xCIM_ComputerSystem>
```

```
  <OperationalStatus>OK</OperationalStatus>
```

```
  <Name>printserver.cs.um.es</Name>
```

```
  <Dedicated>Print</Dedicated>
```

```
</xCIM_ComputerSystem>
```

# SPL examples

- Policy elements:

```
<xCIM_Role>
```

```
  <Name>St_ComputerScience</Name>
```

```
  <BusinessCategory>Students</BusinessCategory>
```

```
</xCIM_Role>
```

```
<xCIM_AuthorizedPrivilege>
```

```
  <InstanceID>PrintAuth</InstanceID>
```

```
  <PrivilegeGranted>>true</PrivilegeGranted>
```

```
  <Activities>Create</Activities>
```

```
</xCIM_AuthorizedPrivilege>
```

# SPL examples – defining the rule

- Policy rule:

```
<xCIM_AuthorizationRule>  
    <PolicyRuleName>PrintRuleAuth</PolicyRuleName>  
</xCIM_AuthorizationRule>
```

- Policy associations:

```
<xCIM_AuthorizationRuleAppliesToTarget>
```

```
    <CIM_AuthorizationRuleAppliesToTarget.AuthorizationRule  
    classReferenced="CIM_AuthorizationRule">[PolicyRuleName=PrintRule  
    Auth] </CIM_AuthorizationRuleAppliesToTarget.AuthorizationRule>
```

```
    <CIM_AuthorizationRuleAppliesToTarget.Element  
    classReferenced="CIM_ComputerSystem">[Name=printserver.cs.um.es]  
    </CIM_AuthorizationRuleAppliesToTarget.Element>
```

```
</xCIM_AuthorizationRuleAppliesToTarget>
```

# SPL examples – defining the rule

## ■ Policy associations:

```
<xCIM_AuthorizationRuleAppliesToRole>
```

```
  <CIM_AuthorizationRuleAppliesToRole.AuthorizationRule
```

```
    classReferenced=
```

```
    „CIM_AuthorizationRule”>[PolicyRuleName=PrintRuleAuth]
```

```
  </ CIM_AuthorizationRuleAppliesToRole.AuthorizationRule>
```

```
  <CIM_AuthorizationRuleAppliesToRole.Role
```

```
    classReferenced=„CIM_Role”>[Name=St_ComputerScience]
```

```
  </ CIM_AuthorizationRuleAppliesToRole.Role>
```

```
</xCIM_AuthorizationRuleAppliesToRole>
```

# SPL examples – defining the rule

- Policy associations:

```
<xCIM_AuthorizationRuleAppliesToPrivilege>
```

```
  <CIM_AuthorizationRuleAppliesToPrivilege.AuthorizationRule  
  classReferenced="CIM_AuthorizationRule">[PolicyRuleName=PrintR  
  uleAuth]
```

```
  </CIM_AuthorizationRuleAppliesToPrivilege.AuthorizationRule>
```

```
  <CIM_AuthorizationRuleAppliesToPrivilege.Privilege  
  classReferenced="CIM_AuthorizedPrivilege">[InstanceID=PrintAuth]
```

```
  </CIM_AuthorizationRuleAppliesToPrivilege.Privilege>
```

```
</xCIM_AuthorizationRuleAppliesToPrivilege>
```

# SDL and SPL consoles

# SDL Console - features

- Validation of SDL with schema
- Transformation of SDL to SDL-CIM
- Validation of SDL-CIM with schema
- Possibility to export and import network descriptions
- Network topology
  - Shows details about the network nodes
  - Enables to find path between network components

# SDL Console

The screenshot displays the SDL Console application interface. At the top, there is a menu bar with options: Import, Export, Transformation, Instances, and Validation. Below the menu bar, there are tabs for SDL, CIM, Instances, Topology, and Options. The main workspace shows a network topology diagram with the following components and connections:

- example\_1.Internet** (red box) is connected to **example\_1.R1** (red box).
- example\_1.a** (cyan box) is connected to **example\_1.R1** (red box).
- example\_1.R1** (red box) is connected to **example\_1.S1** (cyan box).
- example\_1.S1** (cyan box) is connected to **example\_1.client1** (cyan box).
- example\_1.R1** (red box) is also connected to **example\_1.server1** (cyan box).

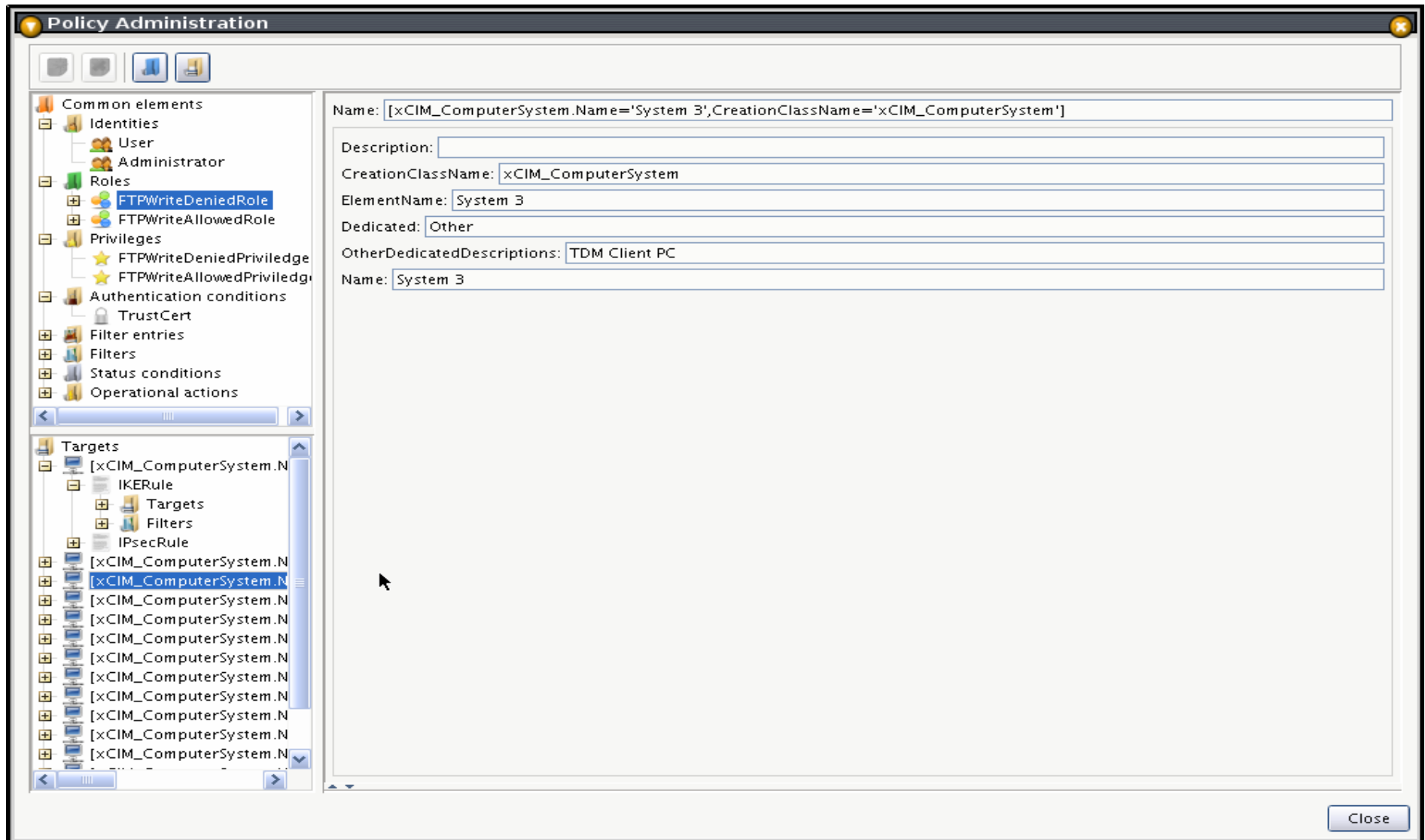
A **Properties** window is open on the right side of the console, displaying the configuration for **example\_1.server1**:

- example\_1.server1
  - Interface example\_1.server1.eth0
    - Address 192.168.1.17/255.255.255.0
    - Address 00:08:02:E7:FF:EA
  - example\_1.MyOS 2.9
    - example\_1.MyOS 2.9
  - example\_1.server1.webserver
    - port 80
  - example\_1.server1.Apache 2.0.1
    - Apache 2.0.1
  - example\_1.server1.ftp server
    - port 21
  - example\_1.server1.Vsftpd 1.3
    - Vsftpd 1.3

# SPL Console - features

- Enables of easy definition of security policies and its components
- Stores policy data in the framework repository

# SPL Console



# SDL & SPL languages and consoles

**Questions...**