



Objectives

- ➔ To design and implement a violation response platform that:
 - ▶ Allows a System Manager to specify response policies for attack scenarios
 - ▶ Defines an interface to Violation Detection to receive notifications of attack violations
 - ▶ Provides a scalable distributed architecture for deploying policies among distributed heterogeneous firewalls and routers

Policy-Based Response

- ➔ Specifies the action(s) to be taken to stop or mitigate an attack, using parameters from the event notification
 - ▶ Perform an action on FE or VD
 - Rate limit the traffic to destined victim
 - Block packets from specific source
 - ▶ Generate and send a new event for the VD or FE PMA
 - ▶ Trigger Traceback to locate the points closest to attack sources

➔ Policies are of the form:

```
on event (params) {
    do managedObject.action
    when condition
}
```

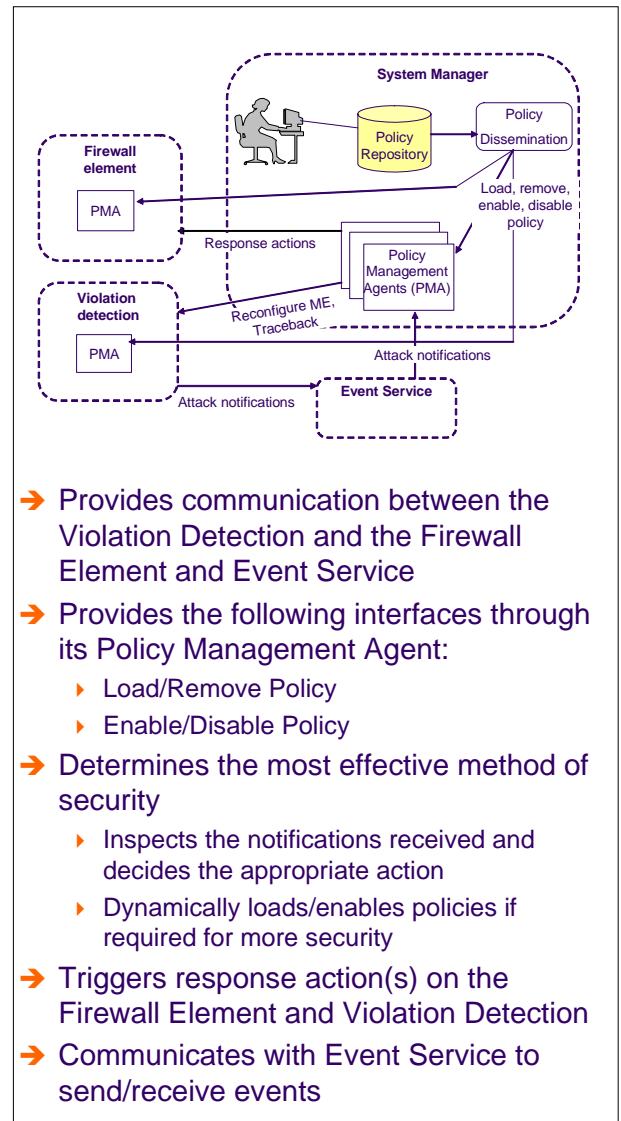
➔ Policies are specified using an XML-based notation

```
<create type="obligation" event="/Event/notification" active="true">
  <action>
    <use name="/Diadem/SystemManager">
      <notification>
        lidmef_msg;
      </notification>
    </use>
  </action>
</create>
```

Integration

- ➔ Integration with Notification Service
 - ▶ Integrated the PMA with xmlBlaster; it subscribes to receive events and publishes events from/to the xmlBlaster server
- ➔ Interface with Firewall Element
 - ▶ The FE provides a common API for implementing response actions on all types of firewall devices (Linux, Cisco, Hardware): rateLimit, redirect, drop, loadModule etc.
- ➔ Integration with Violation Detection/Traceback
 - ▶ On receipt of a TCP SYN flood attack the SM generates a 'StartTraceback' event for the VD. Traceback is performed and the results returned a notification event for the SM

The System Manager (SM)



- ➔ Provides communication between the Violation Detection and the Firewall Element and Event Service
- ➔ Provides the following interfaces through its Policy Management Agent:
 - ▶ Load/Remove Policy
 - ▶ Enable/Disable Policy
- ➔ Determines the most effective method of security
 - ▶ Inspects the notifications received and decides the appropriate action
 - ▶ Dynamically loads/enables policies if required for more security
- ➔ Triggers response action(s) on the Firewall Element and Violation Detection
- ➔ Communicates with Event Service to send/receive events

Performance

- ➔ The SM was used with the two use-cases of the project, testing its performance within multiple scenarios of each use-case
- ➔ The main performance measurements during the tests were:
 - ▶ how quickly the SM arrives at a response decision after receiving the attack notification
 - ▶ how quickly it could trigger an action on the appropriate FE.
- ➔ In all the tests, the average time between receiving a notification of an attack and when the corresponding action is triggered successfully on an FE is approximately **3 seconds**