



SAFE CONFIGURATION AUTOMATION

R. Badonnel, M. Barrère, O. Festor

RESEARCH TEAM
MADYNES

INRIA NANCY GRAND EST

Problem Statement

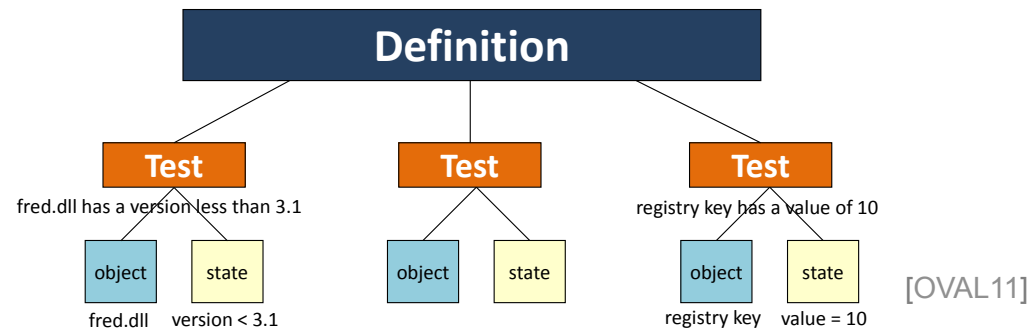
- Growing **complexity** of network and service management (dynamics, scalability)
- **Delegation** of management functionalities to the networks themselves
- Operations and changes executed by autonomic networks may generate **vulnerable states**
- **How can we enable autonomic environments to maintain safe and secure configurations?**

1. Automating Safe Configuration

- Supporting autonomic networks and services
 - Self-* activities: self-healing, self-configuration, self-optimization, self-protection
- Based on vulnerability mgmt techniques
 - Identification, classification and remediation or mitigation of vulnerabilities
 - Unknown vulnerabilities
 - Exp. feedback [Khan08], forensics [Achi08, Cor02]
 - Fuzzing techniques [Dem06, Wang11]
 - Known vulnerabilities
 - Network-based scanning (e.g. Nessus, Nmap)
 - Configuration analysis [OVAL11, AVDL04]

OVAL Language

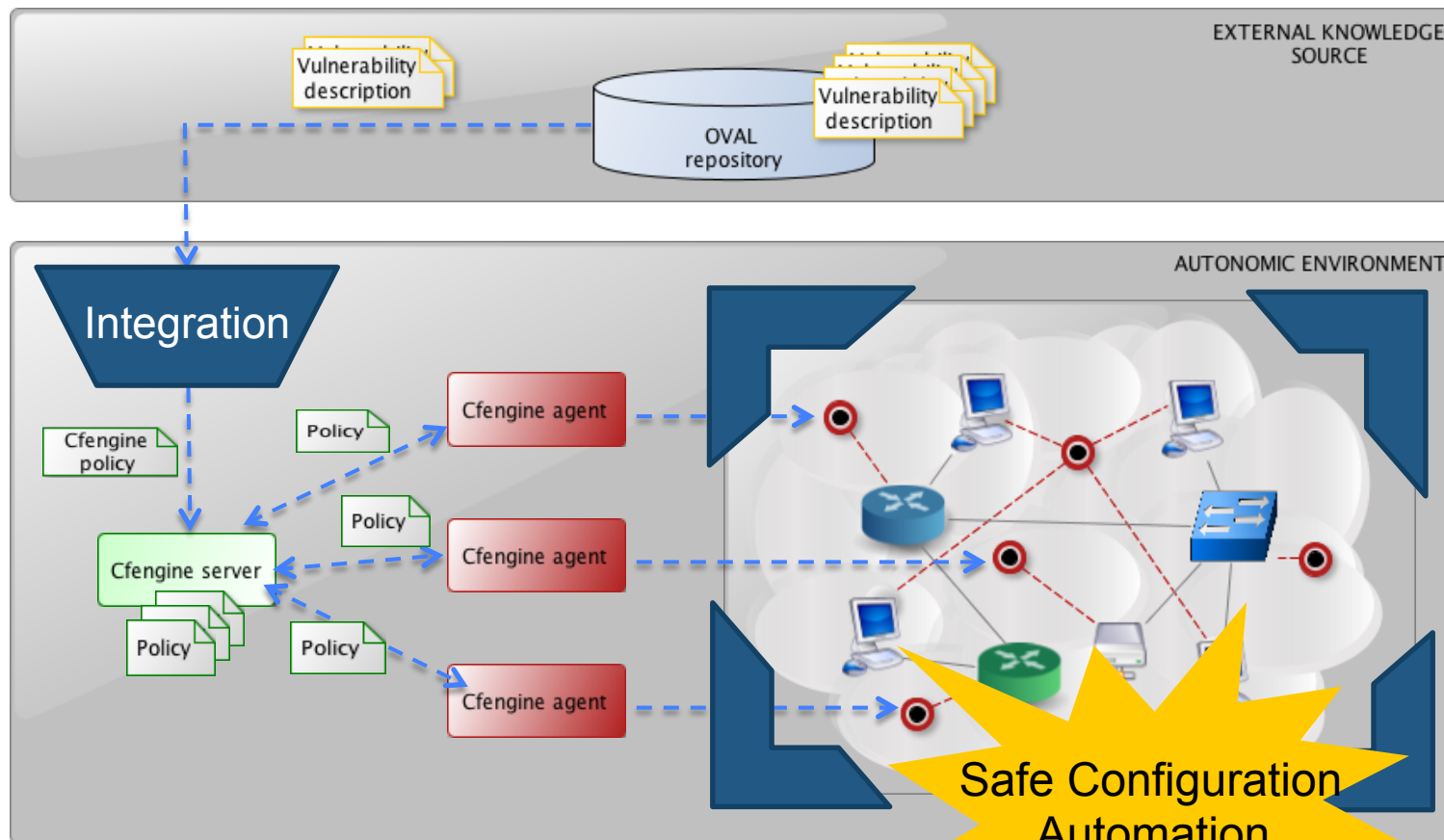
- Open Vulnerability and Assessment Language
 - Part of the SCAP specification (NIST)
- Standardization of vulnerability management information based on three XML schemas
 - Including vulnerability descriptions



- Repository of more than 12500 descriptions
 - <https://oval.mitre.org/repository>

2. Integrating Vulnerability Descriptions into the Mgmt Plane

[CNSM11]



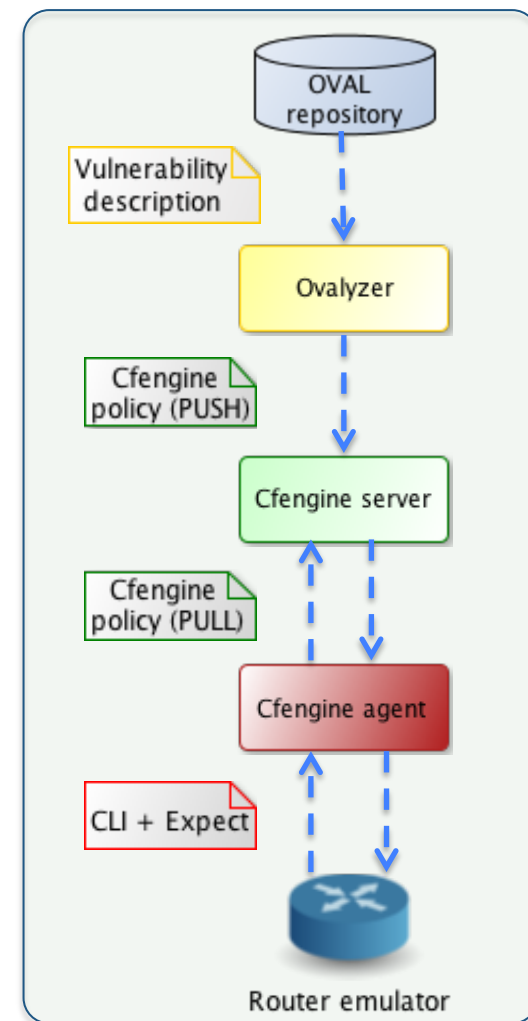
Mapping OVAL to Cfengine Rules

First-Order Logic	OVAL	Cfengine
Atomic predicates	OVAL tests	Cfengine methods
Family of individuals in the discourse universe	OVAL objects	Cfengine prepared modules
Mathematical relationships	OVAL states	Cfengine control variables
Compound logical formulas	OVAL definitions	Cfengine input files
Arrangement of compound logical formulas	OVAL document	Cfengine main configuration file

Cfengine classes are used for expressing results of predicates over the system.

Ovalyzer, an OVAL to Cfengine translator

- Java-based prototype (JAXB, plugins)
- Generation of Cfengine policy rules corresponding to OVAL vulnerability configuration descriptions
- Interpretation of these policy rules by Cfengine running instances for assessing current configurations
- Coverage of IOS official descriptions defined in OVAL v5.8



3. Specifying and Assessing Distributed Configuration Vulnerabilities [NOMS12]

- Network-based scanning (Nessus)
 - **Black-box perspective only** complementary to configuration vulnerability analysis
- Aggregation with XCCDF checklists
 - Collection of (local) host-based vulnerabilities
- Our definition of a distributed vulnerability
 - A set of conditions over two or more network devices that if observed simultaneously, then a potential exploitable flaw is present in the network.
 - **Each device can individually present a secure state!**

Distributed Vulnerability Modelling

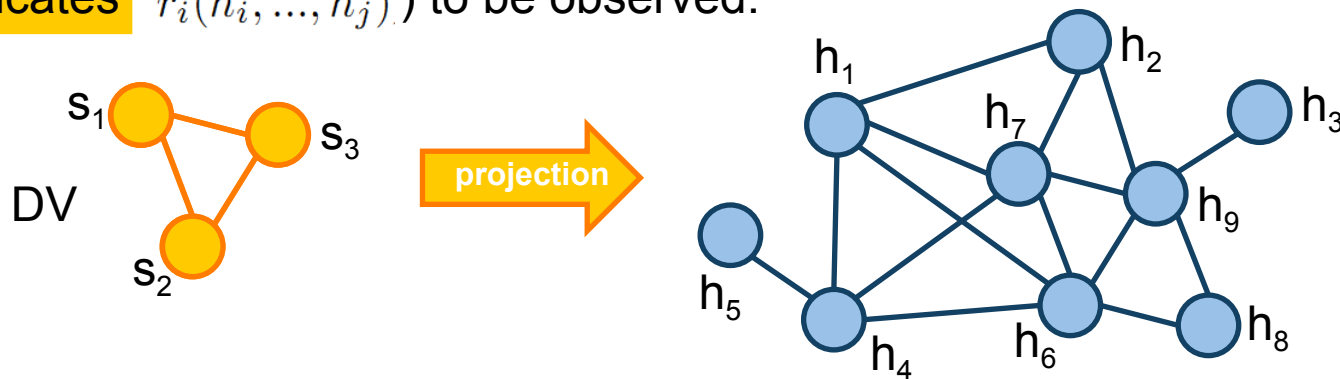
$H = \{h_1, \dots, h_m\}$, set of **devices** in the network (hosts, routers)

$R = \{(h_i, \dots, h_j)*\}$, set of **relationships** amongst devices (reachability, provis.)

$P = \{p_1, \dots, p_n\}$, set of device properties (**unary predicates** $p_i(h_j)$) required,

$P^H = \{s_1, \dots, s_k\}$, set of subsets of P where $s_j = \prod(P) = \{p_i*\}$; s_j is the set of properties, also called *role*, to be observed on one specific device.

$P^R = \{r_1, \dots, r_v\}$, set of relationships amongst network devices (**n-ary predicates** $r_i(h_i, \dots, h_j)$) to be observed.



A distributed vulnerability DV is the compliant projection of the pattern (P^H, P^R) over the network (H, R) .

DOVAL (Distributed OVAL)

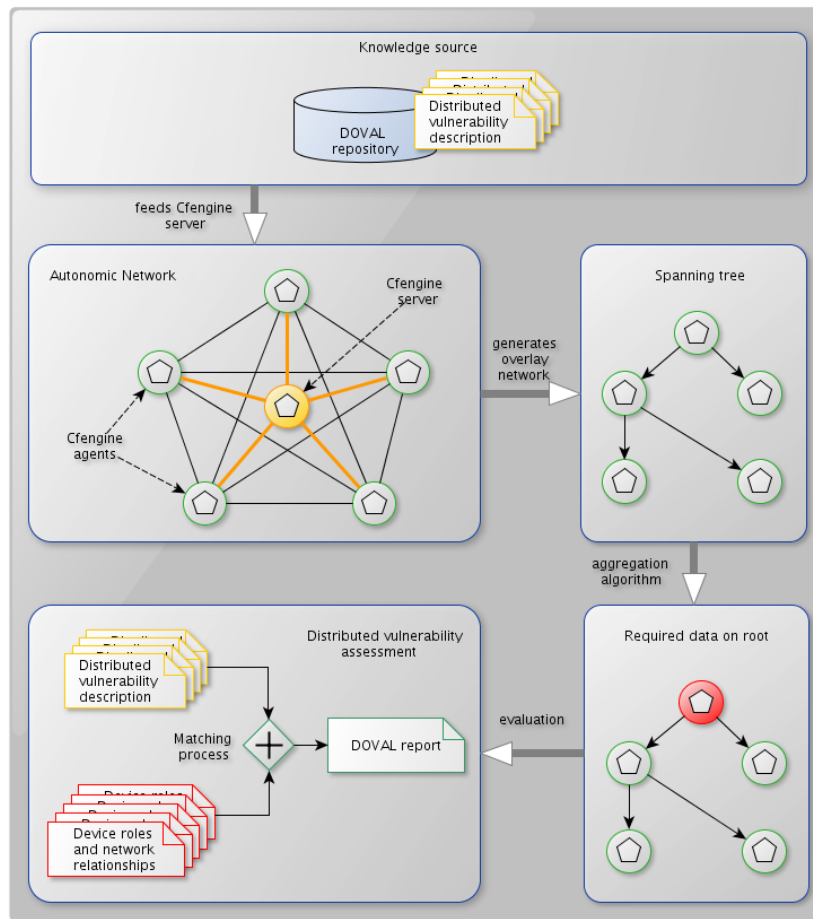
- Definition on top of the OVAL language
 - Expressing predicates of our DV modelling



- Case study in VoIP networks and services
 - Distributed configuration vulnerability spread over the SIP server and the local DNS server

DOVAL Framework

- Storage of DOVAL definitions into repositories
 - Compatibility between DOVAL and OVAL repositories
- Collection of device and network configuration data
 - Cfengine architecture
 - Overlay network with aggregation techniques
- Assessment of DOVAL definitions over gathered data
- Selection of remediation or mitigation actions



Conclusions and Future Work

- Safe configuration automation
 - Integration of OVAL descriptions into the management plane of autonomic networks
 - Modeling and translation algorithm
 - Support architecture based on Cfengine
- Building the DOVAL framework
 - Specification of distributed vulnerabilities
 - Assessment based on optimized aggregations
 - Execution of collaborative corrective actions



SAFE CONFIGURATION AUTOMATION

R. Badonnel, M. Barrère, O. Festor

RESEARCH TEAM
MADYNES

INRIA NANCY GRAND EST