Towards Vulnerability Prevention in Autonomic Networks and Systems

Martin Barrere, Rémi Badonnel, Olivier Festor

Towards Vulnerability Prevention in Autonomic Networks and Systems

Martín Barrère, Rémi Badonnel and Olivier Festor

LORIA - INRIA Nancy Grand Est, France
{barrere, badonnel, festor}@inria.fr

Abstract. The autonomic paradigm has been introduced in order to cope with the growing complexity of management. In that context, autonomic networks and systems are in charge of their own configuration. However, the changes that are operated by these environments may generate vulnerable configurations. In the meantime, a strong standardization effort has been done for specifying the description of configuration vulnerabilities. We propose in this paper an approach for integrating these descriptions into the management plane of autonomic systems in order to ensure safe configurations. We describe the underlying architecture and a set of preliminary results based on the Cfengine configuration tool.

1 Introduction and challenges

The continuous growth and dynamics of networks, as well as the diversification of their services in the context of Future Internet has considerably increased the complexity of their management. In order to face this problem, *autonomic computing* [5], [4], has been introduced providing new perspectives. Highly inspired on the central nervous system, this approach aims to define a strong basis for automated systems capable of managing themselves in an autonomous manner, identifying four major properties, namely, self-configuration, self-optimization, self-healing and self-protection. Despite numerous benefits have been already obtained from this new paradigm, several challenges must be addressed in order to introduce this approach into current systems and networks.

When autonomic related tasks are performed, the environment is modified in order to achieve specific objectives. Such changes may lead to potential vulnerable states, thus change management techniques for assessing change associated risks are required [8]. The vulnerability management activity usually consists in checking the configurations of the system components, identifying the presence of vulnerable states and performing the required maintenance operations (typically, modification of configuration parameters and/or application of security patches). Vulnerability detection and prevention techniques not also increase systems security but also complement the change management process by providing useful information for risk assessment mechanisms.

Even though vulnerability detection techniques have been proposed [7], [3], [6], and mechanisms for uniformly describing vulnerabilities and exchanging related information have been provided [2], there is no integration of such

mechanisms within the framework of autonomic networks and systems. Such integration constitutes the target of our work as we consider that autonomic environments should exploit the knowledge provided by vulnerability repositories in order to increase their security, stability and sustainability.

In this paper we present our approach for integrating vulnerability descriptions in the autonomic management plane, considering the OVAL [2] process and the autonomic maintenance system Cfengine [1]. The remainder of this paper is organized as follows. Section 2 presents the proposed approach for increasing vulnerability awareness within autonomic environments, whereas the results achieved to date are outlined in Section 3. Section 4 presents conclusions and perspectives.

2 Self-configuration with vulnerability prevention

Within the autonomic computing field, the self-configuration property refers to the ability of networks and systems for automatically configuring themselves according to high-level policies. When autonomous networks and systems perform changes in order to be compliant with the specified policies, collateral effects can be introduced without explicit knowledge. Such unexpected effects can vary from internal malfunction to the exposure of vulnerable states.

We propose to support the self-configuration of autonomic systems with vulnerability management mechanisms. These mechanisms can ensure safe configurations and also reduce the probability of potential attacks and failures of the involved self-managed entities. Autonomic systems must be able to perform retro-inspection, identify required changes and execute the appropriate tasks. As happens in the real world, autonomic elements coexist within dynamic environments, interacting with other autonomic and non-autonomic elements. Nevertheless, such scenarios present continuous threats that may compromise autonomic elements safety. If an autonomic element is violated in some way, its functions and abilities become untrustworthy and eventually disabled; thus autonomic elements that use services of the former become compromised as well. This inevitably leads to distrust and the failure of the autonomic system. Autonomic systems must be able to manage their own state and perform the required activities to achieve secure configurations. Autonomic elements unable to support this capability will age with time, becoming more vulnerable, insecure and useless. Automation is really possible only if autonomic networks and systems are capable of ensuring safe configurations.

We therefore argue in favor of the integration of vulnerability descriptions into the management plane of autonomic systems. Our objective is to translate these vulnerability descriptions into policies that are interpretable by an autonomic system. In particular, we propose to translate standardized OVAL[1] vulnerability descriptions into Cfengine policy rules. In this manner, the vulnerability prevention process associated to OVAL can be integrated into Cfengine

[1] Open Vulnerability Assessment Language

devices when maintenance operations are performed as depicted in Figure 1. The OVAL language is a standard XML-based language used by vendors and security organizations for publishing security related information warning about current threats and system vulnerabilities. OVAL repositories offer a wide range of security advisories that can be used for avoiding vulnerable states as well as augmenting networks and systems security considering best practices recommendations. Autonomic maintenance systems such as Cfengine provides support for automating the management of large-scale environments based on high level-policies. Cfengine offers a powerful distributed agent framework that combined with the OVAL vulnerability language, provides an efficient strategy for aligning security aspects on autonomic environments.

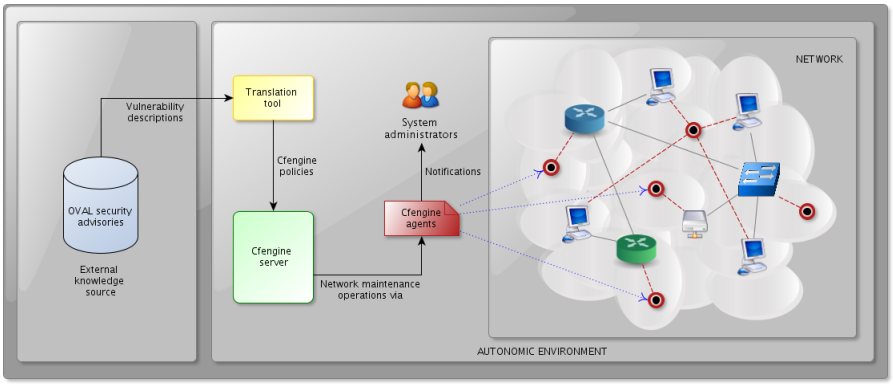

Fig. 1: Self-configuration with vulnerability prevention

In that context, we consider a translation module, identified on Figure 1, responsible for the generation of Cfengine rules corresponding to the vulnerability descriptions defined in the OVAL repository. Generated policies are deployed by the Cfengine server into its several Cfengine agents (points in the cloud) which are in charge of managing the devices present in the target network, in order to detect and prevent vulnerable configurations when self-management activities are performed.

3 Preliminary results

We have formalized our approach by designing an intermediate formal language that provides mathematical tools for supporting the translation between OVAL advisories and Cfengine policies. In order to provide a computable infrastructure for the proposed approach, we have developed a first implementation prototype of the translation engine. At this early stage we have focused on dealing with IOS vulnerabilities over Cisco devices, nevertheless we aim to provide support for managing other platforms as well. The implementation prototype has been designed over a plugin-based architecture with the purpose of enabling easy

means for extending its translation capabilities. Current plugins provide a large variety of IOS vulnerability descriptions available within the official OVAL repository. The logical data model used by the translator is automatically generated based on the OVAL specification, thus enabling a seamless declarative evolution with the OVAL language and providing support for existing and future security related knowledge. We are currently finalizing the proposed prototype and performing several experiments over a simulated environment for evaluating factors such as functionality, performance, and quality of the generated Cfengine code.

4 Conclusions and perspectives

Vulnerability awareness is a key challenge in autonomic networks and systems. The autonomy of such dynamic environments will really be made possible when they fully integrate support mechanisms for preventing vulnerabilities and maintaining safe configurations. In that context, we propose an approach for integrating vulnerability descriptions into the autonomic management plane. We have formalized how these descriptions can be translated into policy rules that are interpretable by an autonomic configuration tool. We have developed a first prototype based on the Cfengine configuration tool, that covers a subset of OVAL definitions and permits to generate vulnerability alerts during the self-configuration activity. For future work, we are interested in extending the coverage of our solution to a larger variety of OVAL definitions, and in investigating further the execution of treatments by the autonomic network, when a vulnerable configuration is observed.

References

[1] Cfengine. `http://www.cfengine.org/`. Last visited on February 14, 2011.

[2] OVAL Language. `http://oval.mitre.org/`. Last visited on February 14, 2011.

[3] H. Achi, A. Hellany, and M. Nagrial. Network Security Approach For Digital Forensics Analysis. *Proceedings of the International Conference on Computer Engineering and Systems (CCES'08)*, pages 263–267, November 2008.

[4] Autonomic Computing. An Architectural Blueprint For Autonomic Computing. *IBM White Paper*, 2006.

[5] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *Computer*, 36(1):41–50, January 2003.

[6] M. J. Khan, M. M. Awais, and S. Shamail. Enabling Self-Configuration in Autonomic Systems Using Case-Based Reasoning with Improved Efficiency. *Proceedings of the 4th International Conference on Autonomic and Autonomous Systems (ICAS'08)*, pages 112–117, March 2008.

[7] T. Wang, T. Wei, G. Gu, and W. Zou. TaintScope: A Checksum-Aware Directed Fuzzing Tool for Automatic Software Vulnerability Detection. *Proceedings of the IEEE Symposium on Security and Privacy*, pages 497–512, May 2010.

[8] J. A. Wickboldt, L. A. Bianchin, and R. C. Lunardi. Improving IT Change Management Processes with Automated Risk Assessment. *Proceedings of the IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'09)*, pages 71–84, 2009.