Security-Configuration Automation for UNIX

Master's Thesis

Supervisor: Prof. Dr. Alexander Pretschner
Advisor: Patrick Stöckle
Email: {alexander.pretschner, patrick.stoeckle}@tum.de
Phone: +49 (89) 289 - 17314
Starting date: immediately

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17314
https://www4.in.tum.de

Context

The standard for machine-readable security-configuration guidelines, SCAP [9], is geared towards automating compliance checks rather than supporting a unified approach for automated implementation and checking. Currently, we at the Chair of Software and Systems Engineering (I4) are developing the Scapolite hardening-framework in cooperation with an industry partner. Recently, we have shown how machine-readable specifications of implementations and checks and their automation can be realized in the context of Windows' based systems. Thus, the logical next step is to expand our approach onto UNIX/Linux based systems. In contrast to the Windows environment, there is one other approach, OpenSCAP [8], which pursues a similar goal. The problem with OpenSCAP is their almost exclusive focus on Red Hat Linux.

Goal

The goal of this thesis is to develop an approach to define machine-readable specifications for UNIX/Linux based systems from which the implementation and the automation of this particular settings can be derived. The difficulty is here that this format has to be as abstract as possible to include as many different distributions as possible, but also as concrete as necessary to be still able to derive the automations. There should be a mechanism to extend the developed approach to assess the peculiarities of single distributions, e.g., through a extension mechanism. Within the UNIX/Linux context, there are many different ways to handle and store configurations, e.g., creating or modifying files or running shell scripts. To simplify this problem, the Augeas framework [5] may be useful. The novel approach should be able to specify all these opportunities and enable the derivation of implementations and checks from them. Furthermore, an extraction method has to be specified to extract the configurations and their values from existing guidelines, e.g., published by the CIS [1]. This extraction method could be implemented similarly to the extraction method we have implemented for the Windows-based guidelines. For the actual execution of the implementations and the checks, it has to be decided, whether we can use preexisting configuration management frameworks like Ansible [7], Chef [3] or Puppet [6]. The alternative is to implement a *library* to apply the implementation or the check of a configuration. The evaluation of this thesis could be conducted during a case study in which the developed approach is applied on a publicly available guideline, e.g., the IASE Red Hat Enterprise Linux guideline [4] or the CIS Ubuntu Linux guideline. On this basis, one could compare the capabilities of the developed approach with OpenSCAP and/or other approaches like the DevSec Hardening Framework [2].

Working Plan

1. Familiarize with the Scapolite framework.
2. Study UNIX/Linux-based guidelines to find similarities and differences.
3. Study other configuration management approaches like Ansible, etc.
4. Create concept for specification of common configuration settings within the Scapolite format.
5. Implement extraction process.
6. Implement derivation of the implementations and the checks.
7. Evaluate the quality of the approach by comparing it to, e.g., OpenSCAP, on the baseline of one or several security-configuration guidelines.

References

[1] Center for Internet Security (CIS). https://www.cisecurity.org/. Accessed: 2019-01-29.

[2] DevSec. https://dev-sec.io. Accessed: 2018-12-18.

[3] Chef Software, Inc. Chef. https://www.chef.io. Accessed: 2019-01-29.

[4] IASE. IASE Red Hat Enterprise Linux 7 STIG Benchmark. Available from https://iasecontent.disa.mil/stigs/zip/U_Red_Hat_Enterprise_Linux_7_V2R2_STIG_SCAP_1-2_Benchmark.zip. Accessed: 2019-01-22.

[5] David Lutterkort. Augeas - A Configuration API. http://augeas.net. Accessed: 2019-02-19.

[6] Puppet, Inc. Puppet. https://puppet.com/. Accessed: 2019-01-29.

[7] Red Hat, Inc. Ansible. https://www.ansible.com/,. Accessed: 2019-01-29.

[8] Red Hat, Inc. OpenSCAP. https://www.open-scap.org, . Accessed: 2018-12-18.

[9] David Waltermire, Stefen Quinn, Harold Booth, and Karen Scarfone. The Technical Specification for the Security Content Automation Protocol (SCAP): Scap version 1.3. Technical report, NIST, 2018. Available from https://csrc.nist.gov/publications/detail/sp/800-126/rev-3/final.

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17314
https://www4.in.tum.de