Cassiopée Project 2018-2019

Aurélien Duboc, Pierrick Gorisse, Lucas Martin
Supervisor: Hervé Debar

Contents

1 Cassiopée Project: Development and deployment of an automated IT security audit tool in a virtualized environment.

Large companies as well as SMEs are subject to a security obligation for their information systems. The idea is to propose a tool that allows the management of the main vulnerabilities that security auditers usually look for. This tool will identify several weaknesses and / or configuration vulnerabilities.
The main interest lies in the automated analysis of a large number of machines. It could also propose automated corrections associated with these weaknesses. One could imagine that a tool like this one, to which other features would be added, could be a security audit equivalency for companies if this tool is accredited.

1.1 Objectives

- The establishment of an active infrastructure resulting from the use of various network services (hypervisor, access control, vpn, dns, reverse proxy, monitoring servers...) as well as the use of personal services and data: some users of this infrastructure host their web and ftp servers.
- The deployment of security audit tools for computer systems running Linux at least.
- The management of the logs associated with these audits.
- The development of a web interface allowing data management and visibility processed by the log management tool.
- The documentation in English for the deliverables and for Github in order to get an easier integration by the open source community.

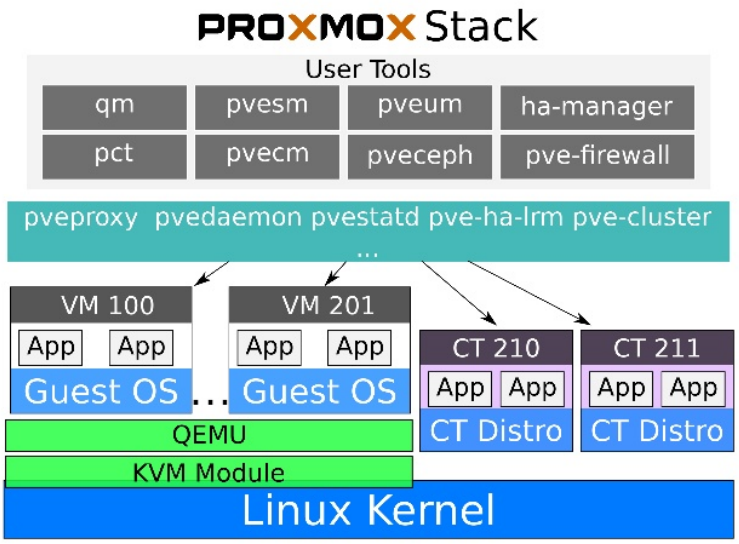
1.2 Requirements

- Setting up a hypervisor for managing virtualized content:

  The chosen solution is Proxmox, an open source hypervisor that can provide container based virtual by way of open VZ. It supports guest operating system like Linux (KVM), Windows. It is enabled by the presence of integrated backup service. It delivers full system virtualization by the use of KVM. The Proxmox management interface can function using a normal browser. Proxmox is using the cluster mode, from a single page multiple servers can be managed. and can perform a direct migration between one host to the other. Lastly, Proxmox provides shell access to the KVM directly from its interface, using a Debian system.

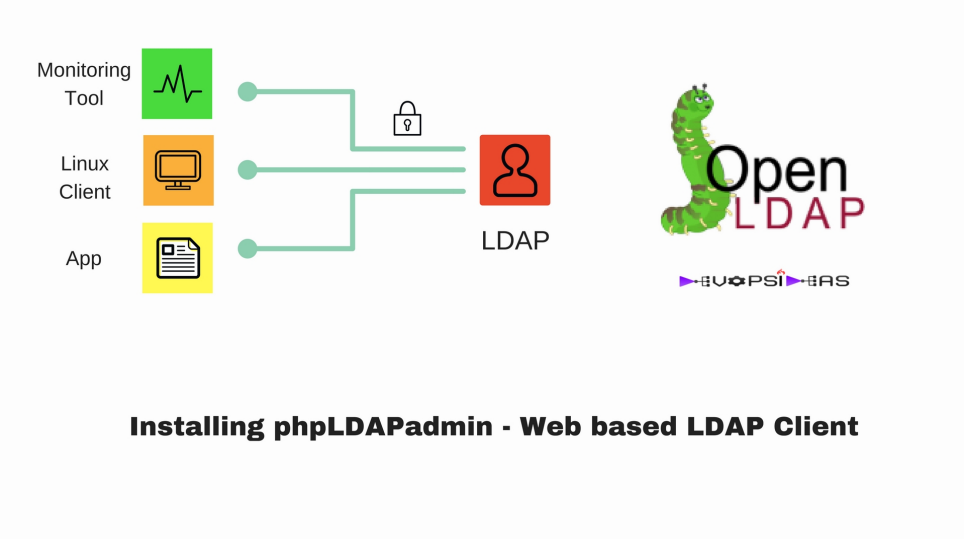
  Proxmox VE tightly integrates KVM hypervisor and LXC containers, software-defined storage and networking functionality on a single platform, and easily manages high availability clusters and disaster recovery tools with the built-in web management interface.

  You may sometimes encounter the term KVM (Kernel-based Virtual Machine). It means that Qemu is running with the support of the virtualization processor extensions, via the Linux KVM module. In the context of Proxmox VE Qemu and KVM can be used interchangeably as Qemu in Proxmox VE will always try to load the KVM module.
  *`https://www.proxmox.com/en/`



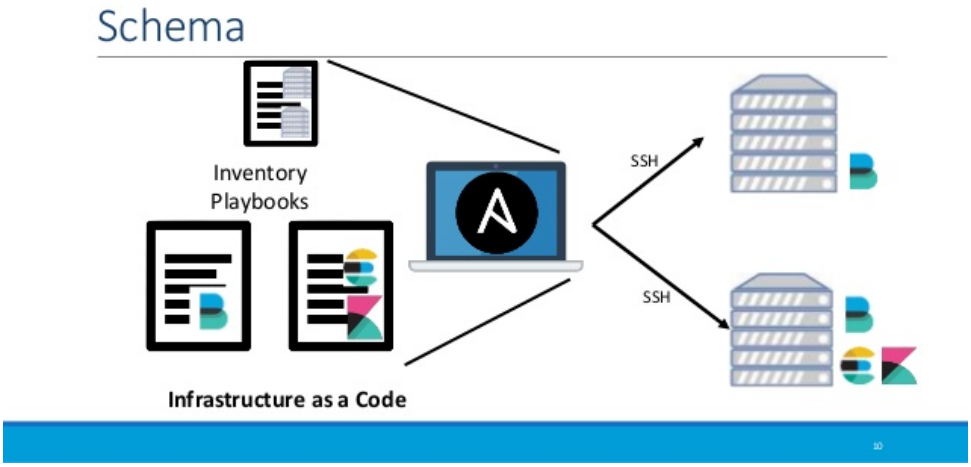
- Setting up a directory to manage users and associated access control policies:

  We are going to use an LDAP directory as specified in RFC 4510 and following. This allows a standardized representation of information (database - LDAP directory) as well as a standard query protocol widely deployed for this database. The chosen solution is the OpenLDAP software version 2.4.46, because of its maturity and protocol compliance. phpLDAPadmin is the web interface that allows the management of user accounts. one of the most important fields is sshPublicKey because all the servers are configured to do LDAP queries during an SSH connection to list the authorized RSA keys. We could have use a PAM setup with libpam-ldap but it seemed easier to specify in ssh configuration files an AuthorizedKeyCommand that will reach all the RSA public keys on the LDAP server.
  *`https://ldap.com/basic-ldap-concepts/`



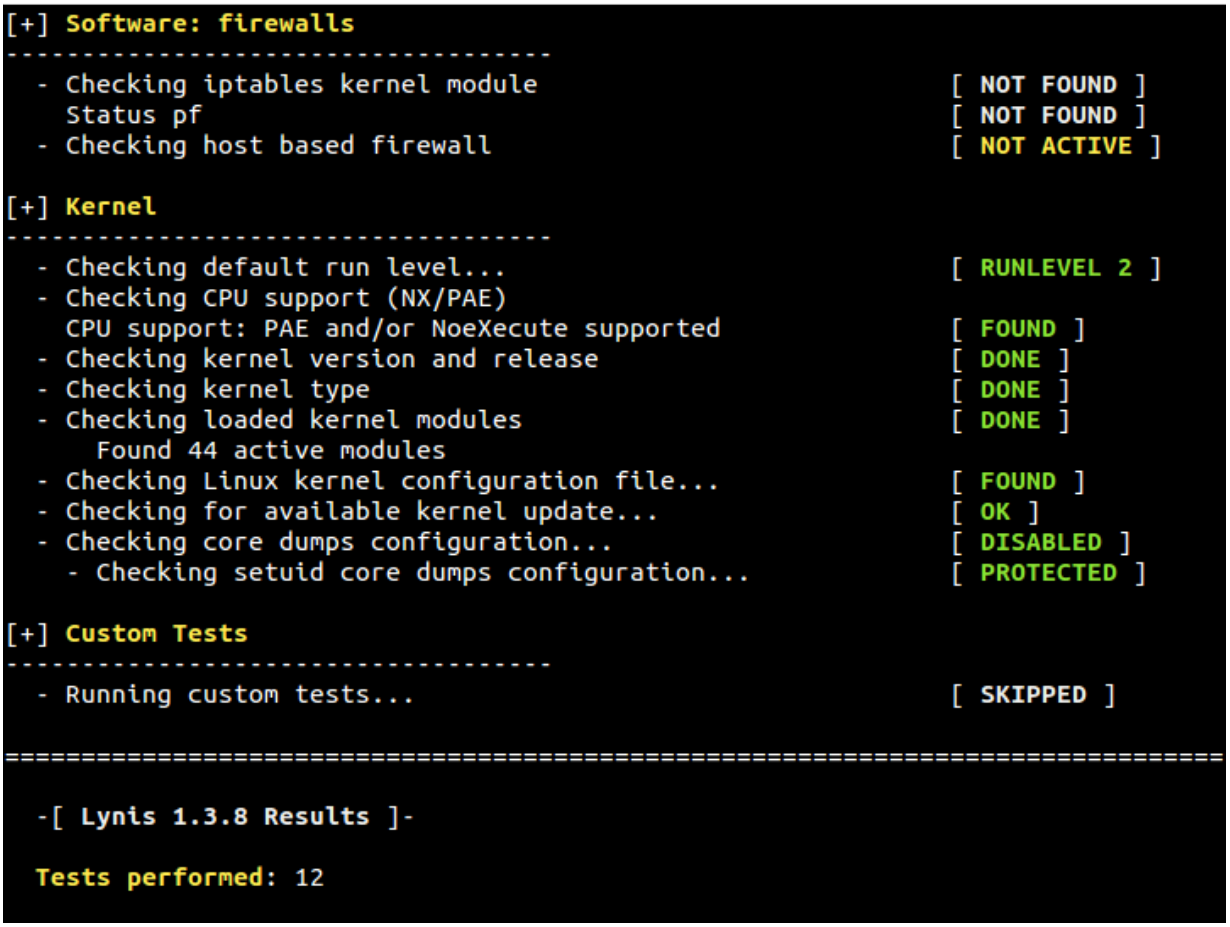
- Use of a tool allowing access to all virtualized machines while using the access control protocol cited above:

  We are going to use Ansible, an open source software that automates software provisioning, configuration management, and application deployment. Ansible connects via SSH, remote PowerShell or via other remote APIs.
  * `(https://www.ansible.com/)`



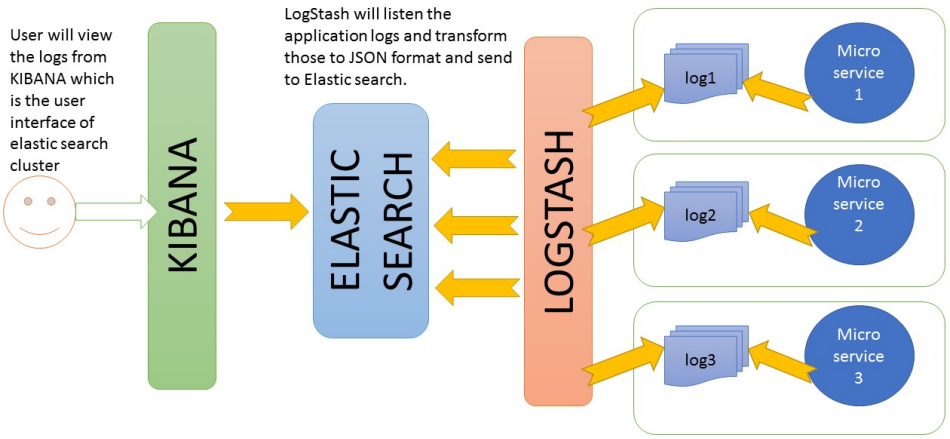
- Implementation of a list of tools allowing the automated audits added to our personal contributions:

  For now, Lynis seems to be the best candidate. We will probably combine several tools later. Lynis is an extensible security audit tool for computer systems running Linux, FreeBSD, macOS, OpenBSD, Solaris, and other Unix-derivatives.
  Lynis scanning is opportunistic, meaning it will only use what it can find, like available tools or libraries. The benefit is that no installation of other tools is needed, so you can keep your systems clean. By using this scanning method, the tool can run with almost no dependencies. Also, the more it finds, the more extensive the audit will be. In other words: Lynis will always perform scans that are customized to your system and two audits will never be the same!
  * `(https://cisofy.com/lynis/)`



- Management of the logs:

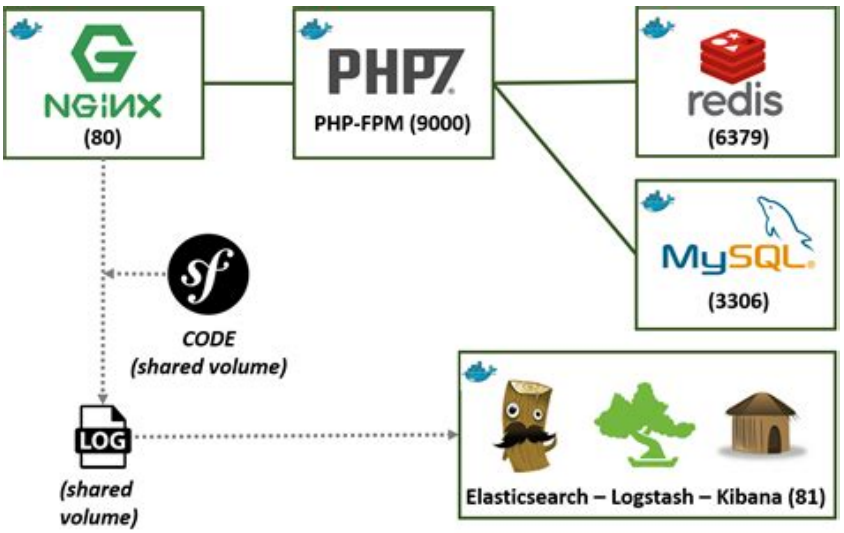
  ELK is the most famous tool for log processing and analysis, so naturally we will use this tool to generate our logs. ELK is the acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a stash like Elasticsearch. Kibana lets users visualize data with charts and graphs in Elasticsearch. The Elastic Stack is the next evolution of the ELK Stack.
  * `(https://www.elastic.co/fr/elk-stack)`



ELK stack interaction with different applications based on Log file

- Web interface development:

  The use of a framework associated with a certain number of libraries will allow us to obtain a modular and easy to use application. Symfony is a PHP framework that we used to manipulate, which is why we chose to use this one. Moreover, we were able to verify that there were many libraries usable by this framework allowing us to interface ELK with our web application.
  * `(https://pehapkari.cz/blog/2017/10/22/connecting-monolog-with-ELK/)`



1.3 Expected results

- A live demonstration is an option, otherwise, we will record a video of our tool performing an automated audit simulation.
- The presentation of the audit log management and administration platform generated by the tool
- To provide a free and open source tool with clear and explicit documentation allowing the redeployment of this tool in a virtualized environment.

We are still looking for other features.

1.4 Project Expectations

The use of English for our project is primodial. Indeed, all the technical documentation associated with the resources used is in English and we are used to working with resources in English because they are much more complete. In addition, the community providing these resources exchanges mainly in English. In order to offer a tool that is widespread and easy to use, writing the documentation in English then appears as a better choice.
Group size: 3 to 4 students.

1.4.1 Specific competencies (in addition to PRO4501):

UNIX-like platforms (Linux, MacOS) and associated tools, including software development tools (editors, interpretors, etc.). We are working on a server without a graphical environment so a text editor like VIM should be optimized as an IDE with some plugins in order to increase our efficiency on the server side, on the development and deployment of the tool. Coding in JavaScript and / or scripting languages, including referenced frameworks for JavaScript development (e.g. Node.js, Angular.js, etc.) As Elasticsearch and Kibana are working as APIs, the frontend development part should work as a REST API too.

1.4.2 Learning objectives

Processing of accessible textual data (coherence management, etc.) , the presentation and analysis of textual data as well as the understanding of cybersecurity software and integration.
Contact: `herve.debar@telecom-sudparis.eu`