CERN

IT-DI-CSO

SUMMER STUDENT 2017

Python-OpenVAS[1]

OpenVAS Vulnerability Scanner CLI wrapper for scan automation & clusterization

Author
Yohan PIPEREAU

Supervisor
Vincent BRILLAULT

September 1, 2017

[1] Program under GPL3 Terms of License and CERN copyright

Introduction

> *OpenVAS is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution.* ——— from www.openvas.org

OpenVAS is working with a set of four main components:

- **openvas-scanner**, providing a scanning service to detect vulnerabilities and output raw results of scans using the OTP[1] protocol;
- **openvas-manager**, providing a manager service to interact with the scanner;
- **openvas-libraries**, providing all libraries for protocols and communication between services;
- **greenbone-security-assistant**, web interface for launching scans.

The initial goal of this project was to build a stable OpenVAS distributed architecture which could be used to distribute scans on different scanning agents using a scan scheduler.

Thus, the first attempt was to use what already existed in the OpenVAS official repository. Unfortunately, because of a choice in OpenVAS developement, a stable clusterisation of OpenVAS was impossible without a wrapper for the scanner. Therefore, I started developping a wrapper being able to talk to OpenVAS scanner using the OTP protocol.

[1] OTP stands for OpenVAS Transfer protocol.It is used for communication between scanner and manager.

System Administration

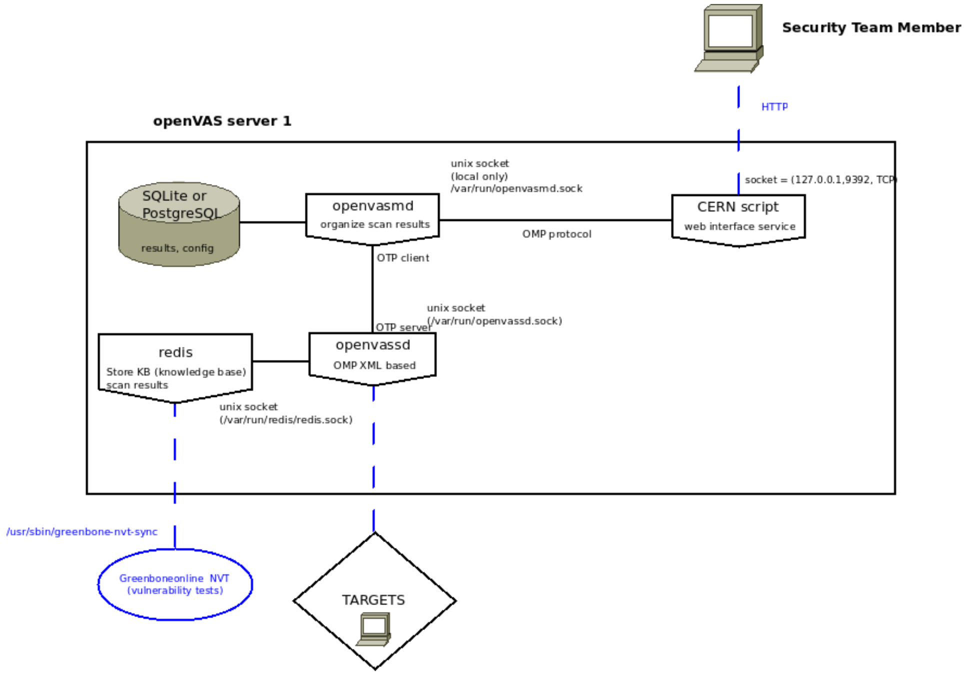

Figure 1: Initial OpenVAS architecture featuring scanner, manager and greenbone-security agent (`gsad`) services

Here is the basic infrastructure installed in the CERN production environment to run OpenVAS scans. As given here, the main problems of the architecture are:

- `gsad` as a web interface is difficult to automate;
- unixsocket used for openvas-scanner service does not allow a single manager to handle multiple scanners. Unfortunately, OpenVAS developpers have chosen to move from TCP sockets to unixsockets which prevents clusterization.

Python-openvas wrapper for OpenVAS scanner

Reversing the OTP protocol tracing syscalls

Though OpenVAS is supposed to be an OpenSource project, documentation is not up-to-date and the source code is C and hardly readable to understand communication between manager and scanner. Thus, the approach I used was using the Linux `strace` tool in order to register the system calls going through the socket. Then, this raw output was parsed in a Python script which gave the OTP syntax it uses.

Human Machine Interface

In order to provide an interface easy to use, to maintain and to execute using a remote scheduler, we needed a command line interface provided in the `python-openvas` file. Then, the Security Team also needed to send report to device owners and members of the team. It also required to keep track of the scan reports i.e. to be saved locally on machines and sent to the CERN Computer Security Operations Centre.

How it works

The tool provides three types of action: retrieving the list of vulnerabilities to scan for, running a scan and sending the scan report in the specified format.

Unix Socket Communication

The unix socket module is the main component of the tool which has evolved a lot to provide stability for running the scans. The client socket communication via `/var/run/openvassd.sock` is entirely dependant on the implementation of the unixsocket server `openvas-scanner`.
The first choice was to use transfer timeout to decide when to end socket interaction. But it proved unstable and unefficient to deal with a buffer size limitation on server side.
I changed the code so that it would not close the socket during vulnerabilities infos retrieval and scan running, moreover I changed the code so that it would try to send a buffer as big as possible and repeat it until the message is fully sent to the socket.

Bug Report: Scanner blocked when overpassing max number of databases

While implementing the tool, the scanner suddenly started to hang while performing a scan and would never finish. I discovered that the `openvas-scanner` did not clean the redis databases it uses if the scan is interrupted by a user. As the number of redis databases is limited scanning was impossible. First, I reported this issue on OpenVAS Mailing List with all the details. Then, it appeared many people faced the same problem, I proposed a workaround which proved successful. Therefore, I created a bash script included inside the python-openvas package which can clean the redis databases in order to fix this bug. You can find in the references, the link to this bug report.

Packaging

python-openvas libraries

In order to allow anyone else to submit an improvement to the OpenVAS project, it has been decided that, the code will be organised as a library with two main programs calling the libraries.
One program is `python-openvas` which provides the interaction with the scanner and the other one is `openvas-blacklist` which allows the user to blacklist vulnerabilities from the scans to avoid meaningless vulnerabilities to be reported.

Package structure

```
python-openvas
 |__ python_openvas
 |    |__ __init__.py
 |    |__ openvas-blacklist
 |    |__ python-openvas
 |    |__ lib
 |         |__ __init__.py
 |         |__ blacklist.py
 |         |__ color.py
 |         |__ iptool.py
 |         |__ oid.py
 |         |__ otp.py
 |         |__ otpsocket.py
 |         |__ parseoid.py
 |         |__ parsescan.py
 |         |__ sendformat.py
 |__ magicredis
 |__ setup.py
```

Conclusion

As a conclusion, the wrapper is working. I managed to run scans targeting VMs I created and to generate reports sent in various format. This project could also proved to be an important OpenSource contribution to the upstream OpenVAS project as it provides a library to communicate using the OTP protocol with the scanner which was really missing to the OpenVAS community.

Last but not least, though I finished this wrapper, there are interesting features which could be deployed, such as:

- Being able to detect RedHat backported packages in reports in order to avoid false-positive caused by backporting security fixes to all packages version;
- Include an OSP[1] wrapper to the project; OSP being the new protocol

[1] OSP stands for OpenVAS Scanner Protocol. It is supposed to become the next generation protocol for communication between manager and scanner.

which will probably substitute OTP in the upcoming years. OSP libraries already exist but are unusable at the moment.

References

CERN Gitlab repository :
`https://gitlab.cern.ch/ComputerSecurity/python-openvas`
Bug reported to the OpenVAS Mailing List :
`http://lists.wald.intevation.org/pipermail/openvas-discuss/2017-August/011380.html`
Help provided to an OpenVAS user on this bug :
`http://lists.wald.intevation.org/pipermail/openvas-discuss/2017-August/011412.html`

Acknowledgement

First, I would like to thank CERN and in particular the Summer Student Program for this wonderful cultural and scientific experience.

And most of all, I am very grateful to the entire security team which has spent time advising me and guiding me in my project. I am in particular very thankful to Vincent Brillault who supervised my project and also to Liviu Valsan who advised me on implementing part of my code.