DEFENSELAYERS ATLAS: CARRYING THE WORLD OF VULNERABILITIES

In Greek mythology, Atlas was one of Titans condemned to hold up celestial heavens for eternity. At Defenselayers Atlas is the main vulnerability database. This technical whitepaper describes atlas role and functionality in Defenselayers secure containers ecosystem.

VULNERABILITY SOURCES

In the containers world there are many sources of vulnerabilities:

- Executable components like web servers or application frameworks
- Shared libraries and other dependencies
- Additional components required to make an application or a microservice work
- Application of microservices' dependencies like Python packages for example

As you can see the list is quite long and complex. Furthermore, for every single component there may be more than a single source of vulnerability information feed. For example single Python package may have multiple CVEs[1] assigned, but at the same time there may be newer package version with no CVE assigned. Furthermore Python package may require binary components like shared library written typically in C/C++ and being a part of the user land operating system. Such dependencies can also have assigned multiple CVEs, however there is always a question if a particular vulnerability had been really patched in the particular revision of the binary component. To solve this issue we use advanced natural language processing algorithms to parse source code tree of the binary components we put into our secure container images. This allows to not only relay on external vulnerabilities databases like NVD[2]. According to our research NVD data is about 10% inaccurate for particular components. Meaning that roughly 1 out of 10 components can be incorrectly classified as secure or insecure. Both cases are dangerous: in the first case a container classified as secure is in fact vulnerable to attack, in the second case perfectly secure container has been classified as insecure and your security team has to waste time to detect so-called false positives. If this happens during your Continuous Integration pipeline one would probably need to change the workflow manually, meaning the processes of building and releasing your software is not automated anymore.

To sum up, atlas processes constantly growing number of data feeders, gathering and classifying vulnerability data in real time, 24/7 and from multiple sources, including:

- External vulnerability databases like NVD
- External source code repositories
- External package management systems for Linux distribution
- External package management systems for particular programming languages

WE KNOW OUR CONTAINERS AND WHAT'S INSIDE THEM

Thanks to our factory module we literally know every bit that we put into every single Defenselayers secure container image. Again, atlas is being used as a intelligent storage system for data gathering during secure images building process within Defenselayers factory module. This allows us to quickly identified possibly vulnerable components when a new vulnerability is discovered. In the future it will also allow to monitor every single secure container image in real time against attacks.

[1] https://cve.mitre.org/
[2] https://nvd.nist.gov/

defenselayers

API OPENING TO EXTERNAL WORLD

Atlas offers also REST API that not only allows all Defenselayers tools to communicate but also opens our databases to external world allowing easier access to data for our customers. It also enables simple on-premise installation of Defenselayers solution for customers who cannot or would not like to use our cloud-based solution.

ANALYTICAL TOOLKIT

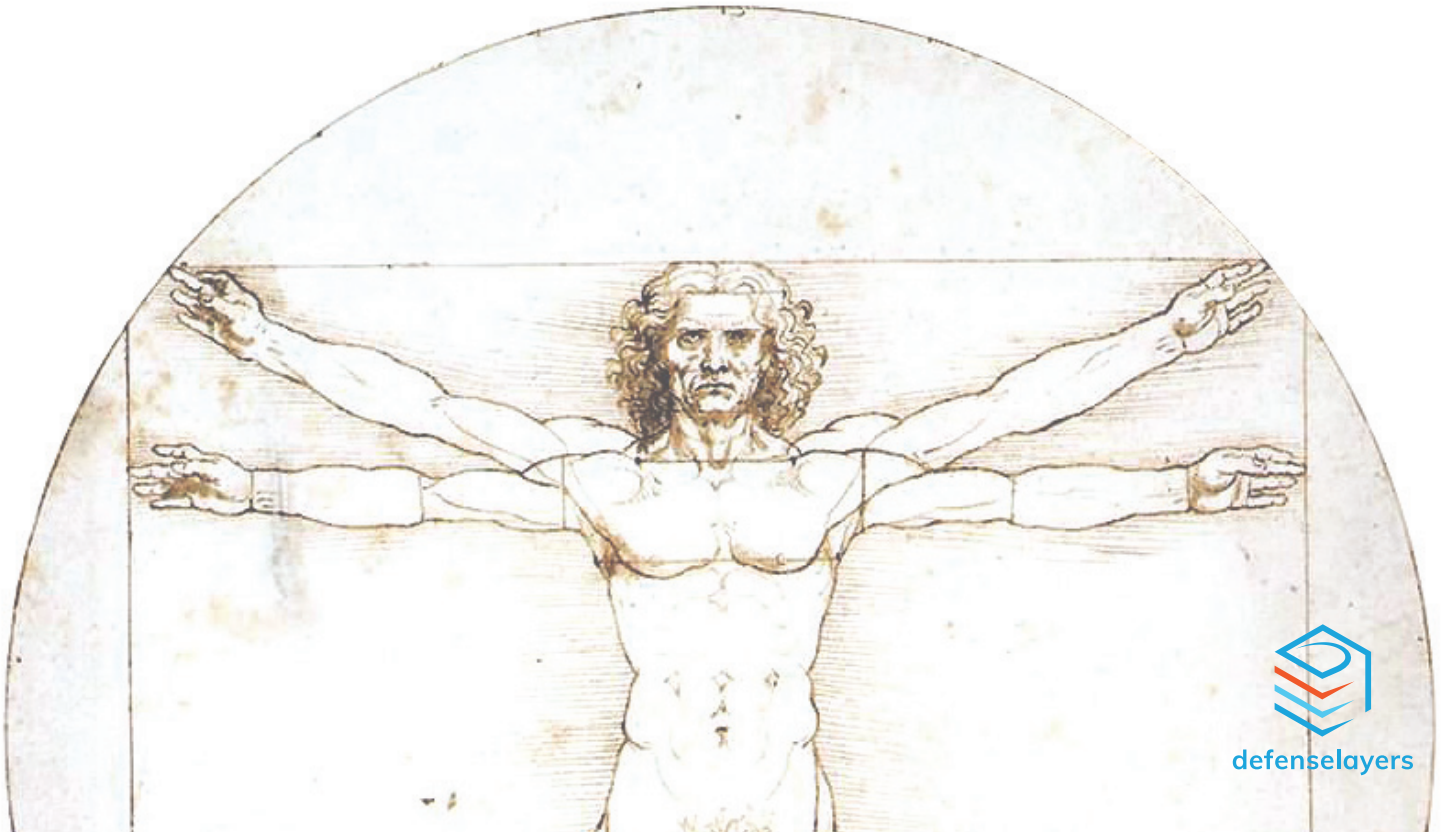
We've build set of innovative analytical tools around atlas for our R&D and Threat Hunting teams. For example, our analytics teams can identify most vulnerable Python package in particular linux distribution with few clicks. With another click it is possible to see how many of the CVEs assigned vulnerabilities are not fixed in a particular component. We are already producing data that is not available publicly. This gives us the important advantage since we know better how secure or insecure particular component one would like to put into container is, then block the malicious or insecure ones.

VULNERABILITY REPORTING ENDPOINT

Thanks to atlas architecture and REST API we provide vulnerability reporting endpoints that enable our customers to gain latest reports in real time about vulnerable components.

ATLAS ARCHITECTURE

Data feeders	REST API	Data model	Analitics
▶ Vulnerability db feeders ▶ Source code feeders ▶ Others	▶ Authentication ▶ Database quries ▶ Reporting	▶ Data storage	▶ Analysis tools



defenselayers