DockerPolicyModules: Mandatory Access Control for Docker Containers

Enrico Bacis, Simone Mutti, Steven Capelli, Stefano Paraboschi
DIGIP — Università degli Studi di Bergamo, Italy
{enrico.bacis, simone.mutti, steven.capelli, parabosc} @ unibg.it

Objectives

We propose an extension to the *Dockerfile* format to let Docker image maintainers ship a specific **SELinux** policy for the processes that run inside the image, enhancing the security of containers.

SELinux Docker Security

Docker leverages Linux kernel security facilities such as Mandatory Access Control (e.g. SELinux). SELinux separates processes in two ways:

- **Type Enforcement**: a type is associated with every process and file. The policy defines the permitted actions among them.
- **Multi-Category Security**: Different containers are assigned different categories to specialize SELinux types.

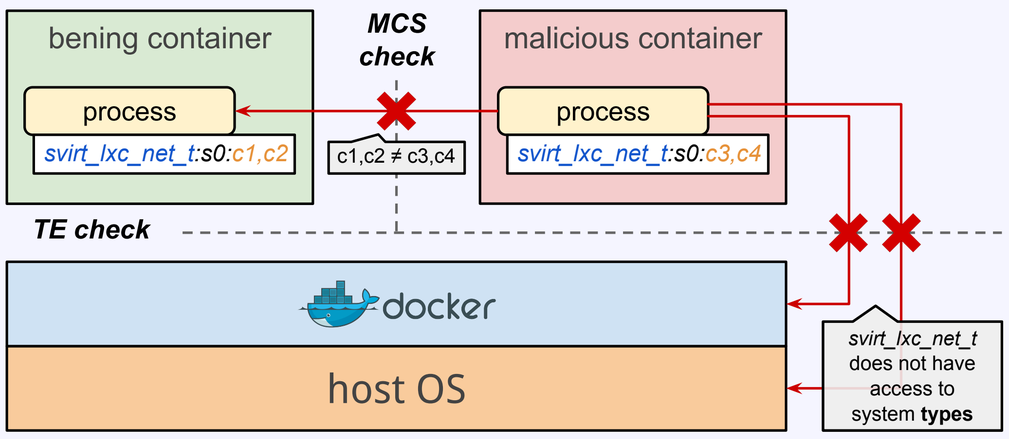

Figure: SELinux separates containers using categories and protect the host through types.

Limitations of the current solution

Currently **all the containers run with the same SELinux type**, *svirt_lxc_net_t*. So we have to grant that type the **upper bound of the privileges** that a container could ever need.

Proposal

Our proposal leverages SELinux modules to allow Docker image maintainers to ship an SELinux policy in conjunction with their images. These modules are named **DockerPolicyModules** (**DPM**) and are used to:

- define the SELinux types and rules for the image;
- define the SELinux type used when starting a containerized process;
- let Docker embed the SELinux policy in the metadata at build-time.

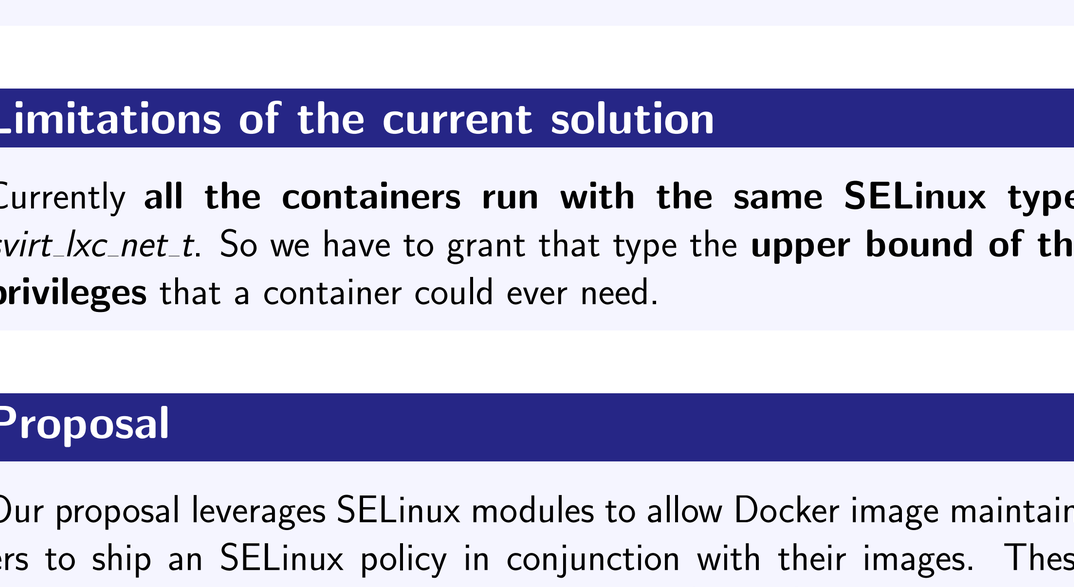

Figure: Process in a *Docker* container with a custom SELinux type defined in the DPM.

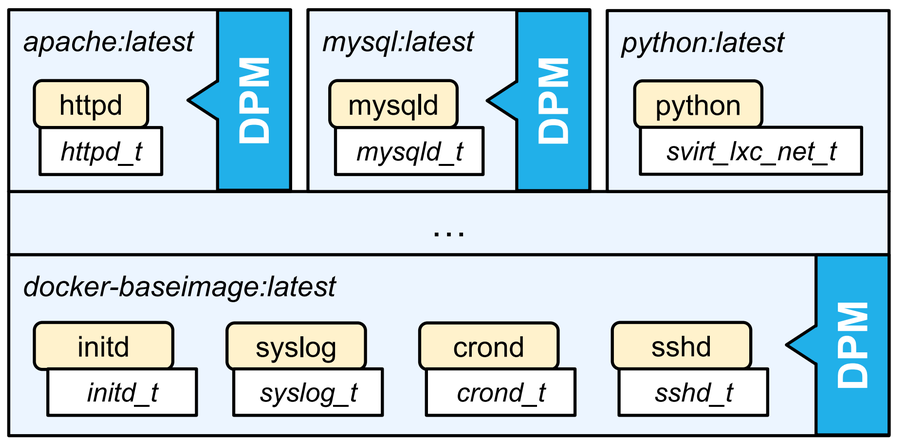

Figure: Processes running in three Docker containers (*apache*, *mysql* and *python*), using SELinux types defined in the DockerPolicyModules embedded in the images.

DockerPolicyModule Validation

Each SELinux rule has a source (σ) and a target (τ) type. They can be defined either in the system policy or in the DPM. We have to check all the cases to avoid possible threats arising from malicious DPMs:

	$\tau \in BASE$	$\tau \in DPM$
$\sigma \in BASE$	**INVALID**. The DPM must not change the types defined in the **system policy**.	**OK / INVALID**. The *typebounds* rule confines the DPM under *svirt_lxc_net_t*.
$\sigma \in DPM$	**OK / INVALID**. The *typebounds* rule confines the DPM under *svirt_lxc_net_t*.	**OK**. Multiple types can be defined with different privileges (*least privilege principle*).

Docker Hub

Docker Hub is an online repository for Docker images. This must ensure that the DPM satisfies the requirements in the table above. The requirements are also verified when Docker downloads the image.

Conclusion

The use of **DockerPolicyModules** permits the specification of specific SELinux types and rules for the processes running in containers, increasing the overall Docker security.

References

Enrico Bacis, Simone Mutti, and Stefano Paraboschi.
AppPolicyModules: Mandatory Access Control for Third-Party Apps.
In *AsiaCCS'15*. ACM, 2015.

Simone Mutti, Enrico Bacis, and Stefano Paraboschi.
Policy Specialization to Support Domain Isolation.
In *SafeConfig'15*. ACM, 2015.

Daniel J Walsh.
Tuning Docker with the newest security enhancements.
In *opensource.com*, 2015.

This work was partially supported by a Google Research Award (winter 2014), by the Italian Ministry of Research within the PRIN project "GenData 2020" and by the EC within the 7FP and H2020 program, respectively, under projects PoSecCo (257129) and EscudoCloud (644579).