



Sysdig

20 Docker Security Tools Compared.

WHITE PAPER



Is Docker insecure?

Not at all. Actually features like process isolation with user namespaces, resource encapsulation with cgroups, immutable images and shipping the minimal software and dependencies reduce the attack vector providing a great deal of protection. But, is there anything else we can do? There is much more than image vulnerability scanning and these are 20 container and Docker specific security tools that can help.

Index of Docker Security tools

[Anchore Navigator](#)

[AppArmor](#)

[AquaSec](#)

[BlackDuck Docker security](#)

[Cavirin](#)

[Cilium](#)

[CoreOS Clair](#)

[Docker capabilities & resource quotas](#)

[Docker-bench security](#)

[Dockscan](#)

[Falco](#)

[HashiCorp Vault](#)

[NeuVector](#)

[Notary](#)

[OpenSCAP](#)

[REMnux](#)

[SELinux](#)

[Seccomp](#)

[StackRox](#)

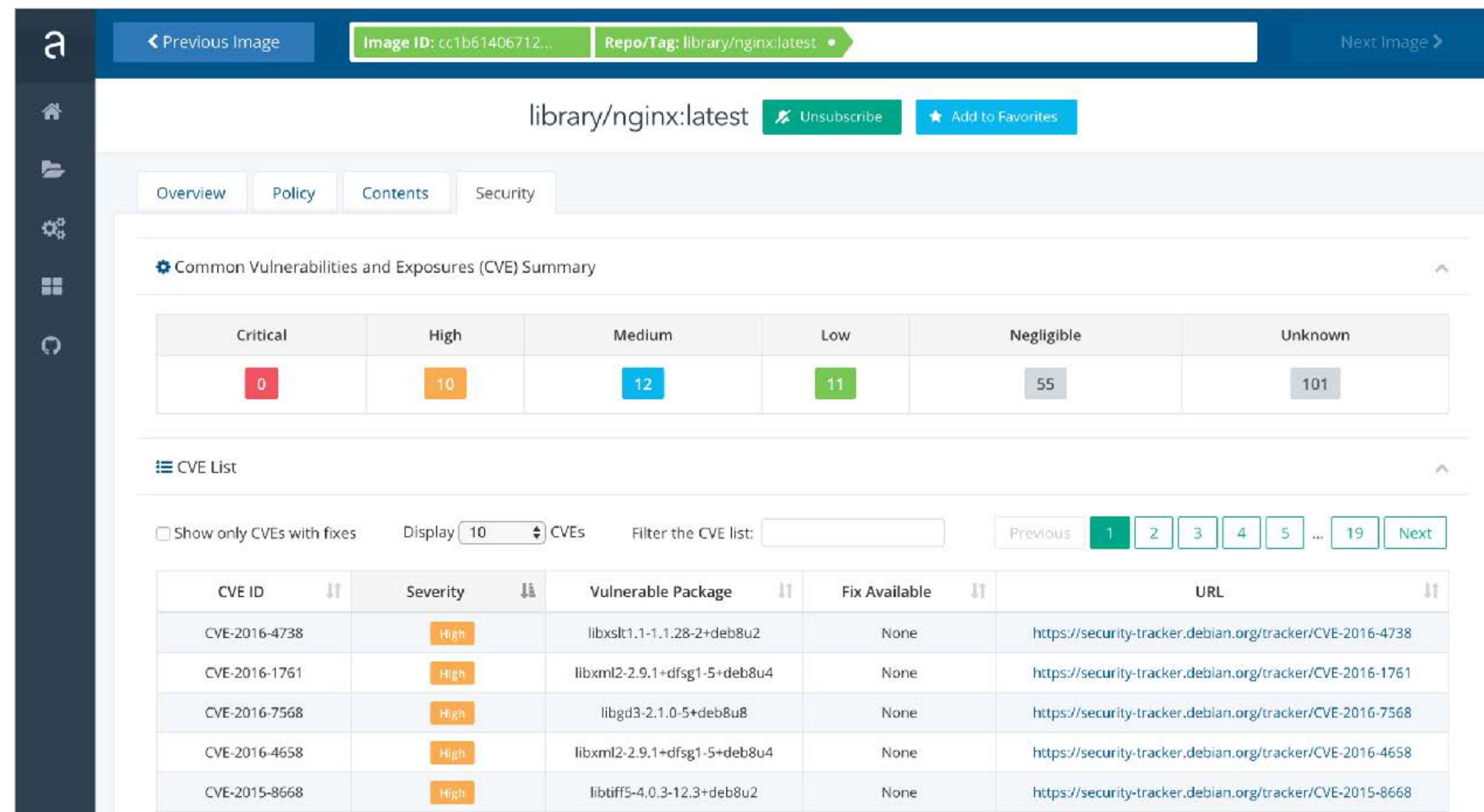
[Sysdig](#)

[Sysdig Secure](#)

[Tenable Flawcheck](#)

[Twistlock](#)

Anchore Navigator



Homepage: anchore.io

License: Commercial, some services are free to use

Use Cases: Pre-production analysis,
vulnerability newsfeed

Anchore Navigator provides a free service for deep inspection of public Docker images. You can also explore their rich repository of dissected public images for full visibility of its content, build process, and discovered CVE threats together with a link with the issue complete description and known fixes.

Using this tool you can perform a deep analysis of your own images and subscribe to the images you frequently use for your deployments to receive security warnings when upgrading to the commercial version.

AppArmor

```
# vim:syntax=apparmor
# Last Modified: Fri Jul 17 11:46:19 2009
# Author: Jamie Strandboge <jamie@canonical.com>
#include <tunables/global>

/sbin/dhclient flags=(attach_disconnected) {
#include <abstractions/base>
#include <abstractions/nameservice>
#include <abstractions/openssl>

capability net_bind_service,
capability net_raw,
capability sys_module,
capability dac_override,
capability net_admin,

network packet,
network raw,
```

Homepage: wiki.apparmor.net

License: Open Source

Use Cases: Runtime protection,
Mandatory Access Control (MAC)

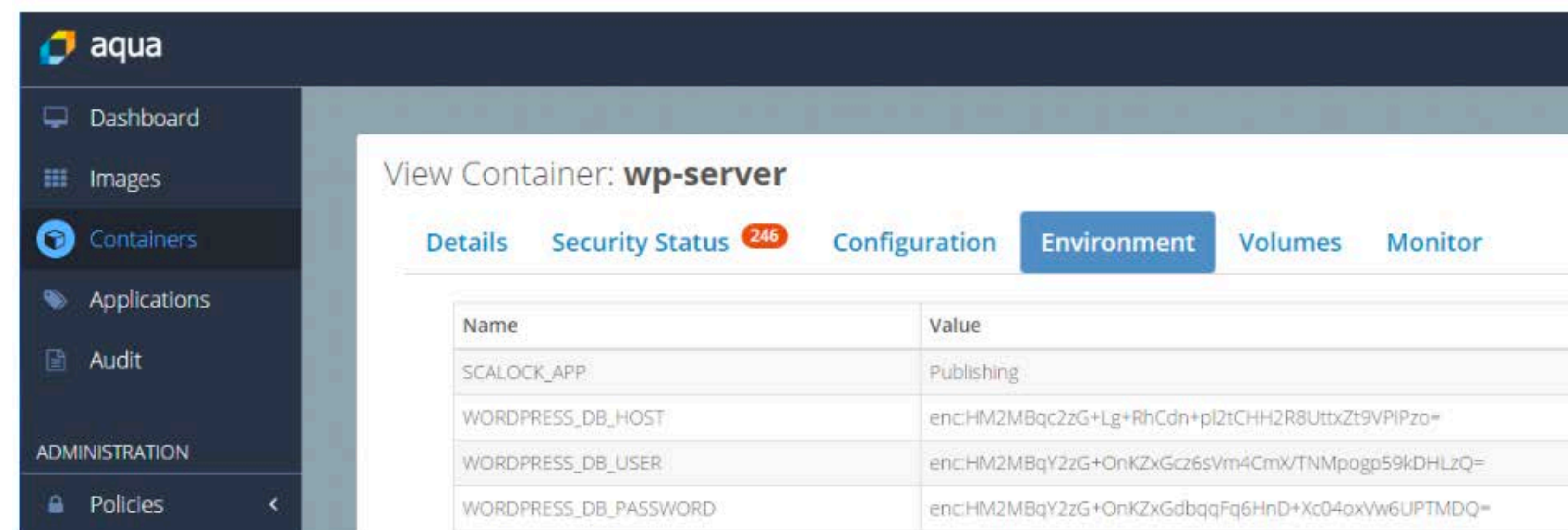
AppArmor lets the administrator assign a security profile to each program in your system: filesystem access, network capabilities, link and execute rules, etc.

It's a Mandatory Access Control (or MAC) system, meaning that it will prevent the forbidden action from taking place, although it can also report profile violation attempts.

AppArmor it's sometimes considered a more accessible and simplified version of [SELinux](#), both are closely related. You only need to learn the [profile language syntax](#) and fire your favorite editor to start writing your own AppArmor rules.

Docker context: Docker can automatically generate and load a default AppArmor profile for containers named docker-default. You can create [specific security profiles](#) for your containers or the applications inside them.

AquaSec



Homepage: aquasec.com

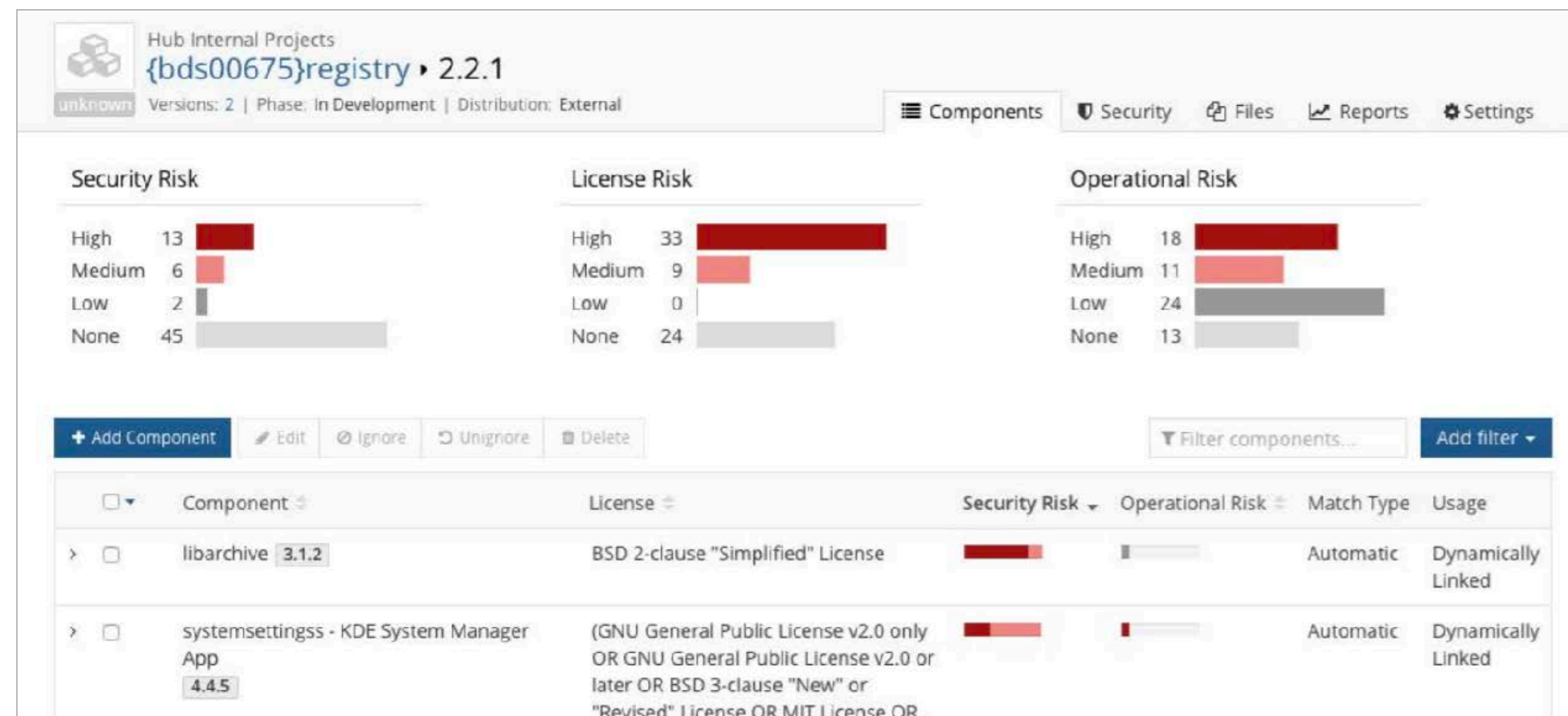
License: Commercial

Use Cases: Pre-production analysis, runtime protection, compliance & audit, etc.

AquaSec is a commercial security suite designed with containers in mind. Security audit, container image verification, runtime protection, automated policy learning or intrusion prevention capabilities are some of the most relevant features.

AquaSec supports orchestration tools like Docker Swarm, Mesos, Kubernetes or OpenShift. The platform provides programmatic access to its API and can be deployed both locally or in the public cloud.

BlackDuck Docker Security



Black Duck Hub specializes in container inventory and reporting image inventory, mapping known security vulnerabilities to images indexes and cross project risk reports. You can easily pinpoint the specific libraries, software packages or binaries that are causing the security risk and the assistant will automatically offer you a list of known fixes.

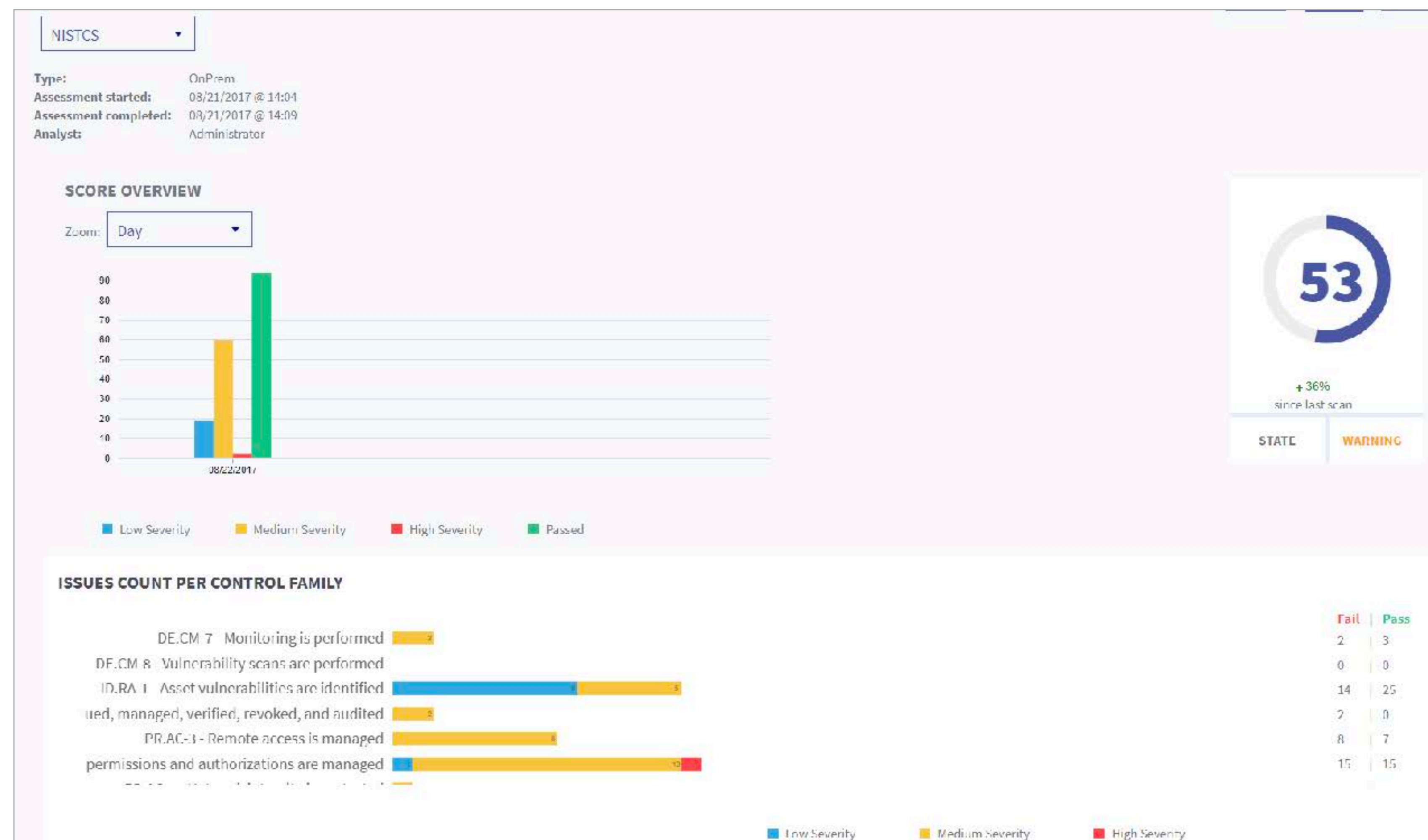
As opposed to similar solutions, Black Duck Hub also analyzes the "License Risk" considering the different software licences that you are currently bundling together to build your containerized distributed system.

Homepage: blackducksoftware.com

License: Commercial.

Use Cases: Pre-production analysis, vulnerability newsfeed, license/legal risks.

Cavirin



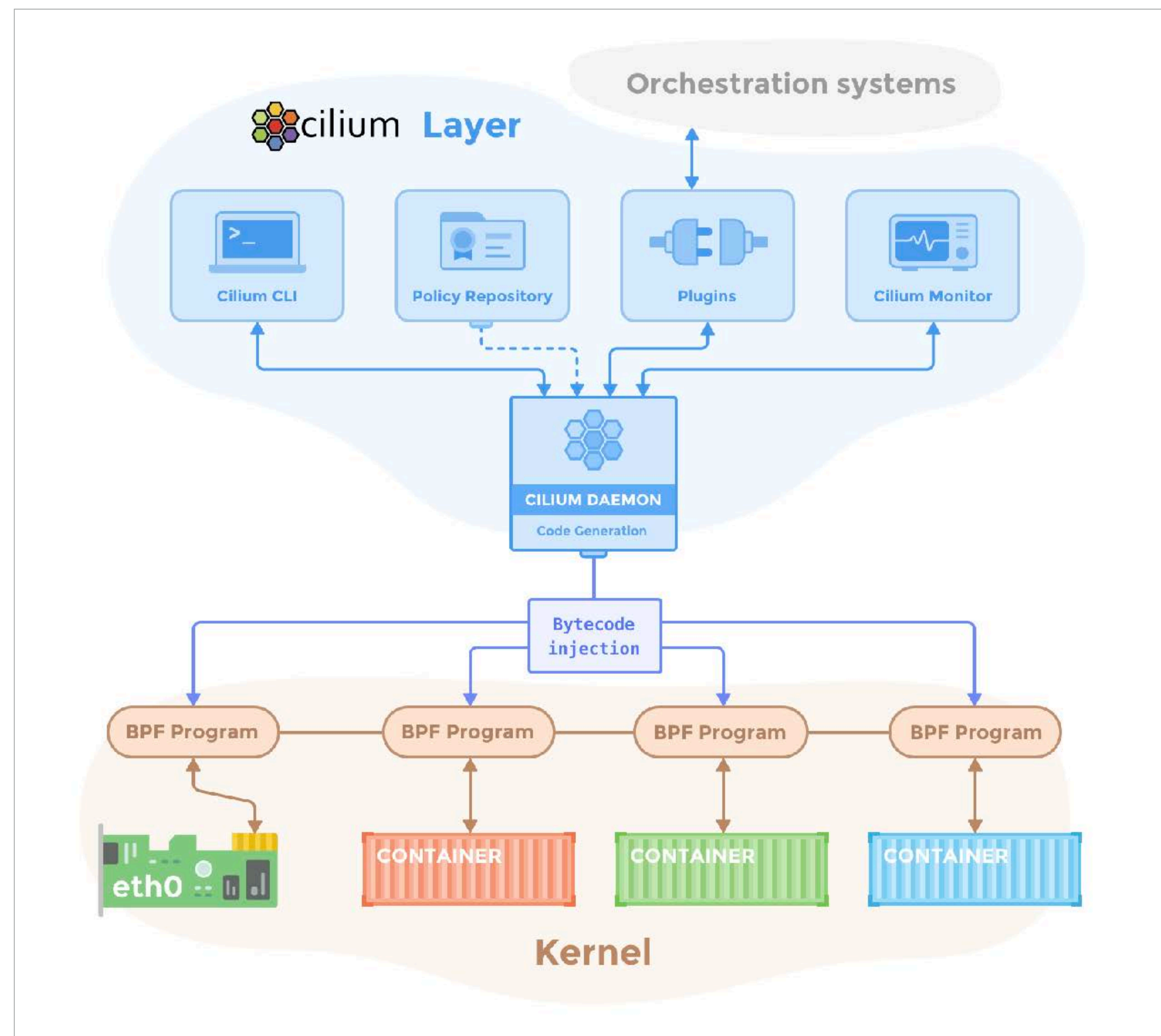
Cavirin works with organizations such as CIS to collaboratively develop and maintain the security standards that any other tool can benefit from. At present, it has authored CIS Docker Security Benchmark as well as CIS Kubernetes Security Benchmark. They have minted the term "DevSecOps" to stress their focus at integrating the security and DevOps/container fields. Apart from the features you can expect in a one-stop DevOps security platform (maybe comparable to Twistlock or AquaSec in their feature proposal and approach), we can highlight their compliance&audit tooling for security standards like PCI, HIPAA, NIST or GDPR.

Homepage: cavirin.com

License: Commercial

Use Cases: Runtime protection, pre-production analysis, compliance & audit

Cilium



Cilium provides transparent network security between container applications. Based on a new Linux kernel technology called eBPF, allows to define and enforce both network-layer and HTTP-layer security policies based on container/pod identity.

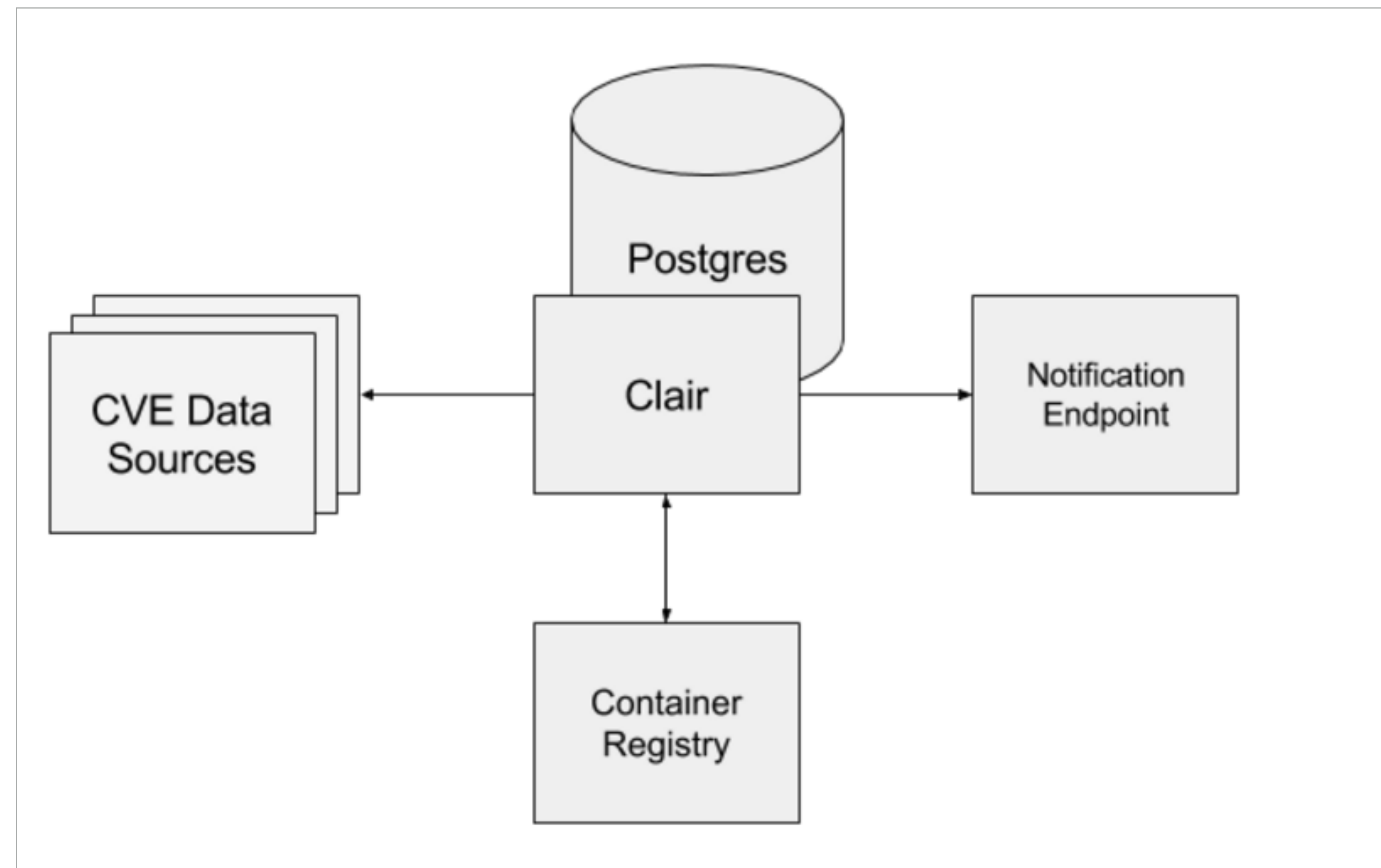
Cilium leverages BPF to perform core data path filtering, mangling, monitoring and redirection. These BPF capabilities are available in any Linux kernel version 4.8.0 or newer.

Homepage: cilium.io

License: Open Source

Use Cases: HTTP-layer network security,
network-layer security

CoreOS Clair



Clair is an open source project for the static analysis of vulnerabilities in containers (currently supporting AppC and Docker). Clair periodically refreshes its vulnerability database from a set of configured CVE sources, scrubs the available container images and indexes the installed software packages. If any insecure software is detected, it can alert or block deployment to production.

Since Clair image analysis is static, containers never need to be actually executed, so you can detect a security threat before is already running in your systems. Clair is the security engine used internally for the CoreOS Quay container registry.

Homepage: coreos.com/clair

License: Open Source

Use Cases: Pre-production analysis, vulnerability newsfeed

Docker capabilities and resource quotas

```
$ sudo docker build -t cpu-stress .
Sending build context to Docker daemon 3.072 kB
Step 1 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
90d6565b970a: Pull complete
40553bdb8474: Pull complete
<snip>
Step 3 : CMD stress -c 2
---> Running in b90defccbb8
---> 9e6a3f316e91
Removing intermediate container b90defccbb8
Successfully built 9e6a3f316e91
```

Homepage: docker.com

License: Open Source

Use Cases: Runtime protection,
resource DoS protection

We shouldn't forget the basic security measures that come already bundled with our OS and the Docker engine.

Resource abuse and denial of service is an often overlooked but very real security problem in a containerized environment with vast amounts of software entities competing for the host resources.

Control Groups (cgroups) is a feature of the Linux kernel that allows you to limit the access processes and containers have to system resources such as CPU, RAM, IOPS and network.

Capabilities allows you to break down the full root permissions into several split permissions, this way you can remove specific capabilities from the root account or augment the capabilities of user accounts at a more granular level.

Docker-bench security

```
Initializing Fri Jul 14 09:18:42 UTC 2017

[INFO] 1 - Host Configuration
[WARN] 1.1 - Ensure a separate partition for containers has been created
[NOTE] 1.2 - Ensure the container host has been Hardened
[PASS] 1.3 - Ensure Docker is up to date
[INFO] * Using 17.06.0 which is current
[INFO] * Check with your operating system vendor for support and security maintenance for Docker
[INFO] 1.4 - Ensure only trusted users are allowed to control Docker daemon
[INFO] * docker:x:992:vagrant
[WARN] 1.5 - Ensure auditing is configured for the Docker daemon
[WARN] 1.6 - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.7 - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.8 - Ensure auditing is configured for Docker files and directories - docker.service
[INFO] 1.9 - Ensure auditing is configured for Docker files and directories - docker.socket
[INFO] * File not found
[INFO] 1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO] * File not found
[INFO] 1.11 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[INFO] * File not found
[WARN] 1.12 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd
[WARN] 1.13 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-runc
```

The Docker Bench for Security is a meta-script that checks for dozens of common best-practices around deploying Docker containers in production.

This script is conveniently packaged as a Docker container, just copying and pasting the docker run one-liner from its homepage you can instantly see the results of ~250 checks for your running Docker containers and the host running the Docker engine (Docker CE or Docker Swarm). Docker Bench tests are inspired by the [CIS Docker Community Edition Benchmark v1.1.0](#)

Homepage: github.com/docker/docker-bench-security

License: Open Source

Use Cases: Compliance & audit

Dockscan

```
root@zaphod:~# dockscan -v
I, [2017-07-24T23:20:47.522728 #833] INFO -- : Validating version specified: unix:///var/run/docker.sock
I, [2017-07-24T23:20:47.554769 #833] INFO -- : Loading discovery modules...
I, [2017-07-24T23:20:47.555167 #833] INFO -- : Loading discovery module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/over/get-containers.rb
I, [2017-07-24T23:20:47.555656 #833] INFO -- : Loading discovery module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/over/get-docker-info.rb
I, [2017-07-24T23:20:47.556047 #833] INFO -- : Loading discovery module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/over/get-run-containers.rb
I, [2017-07-24T23:20:47.556452 #833] INFO -- : Loading discovery module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/over/get-docker-version.rb
I, [2017-07-24T23:20:47.556713 #833] INFO -- : Loading discovery module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/over/get-images.rb
I, [2017-07-24T23:20:47.557015 #833] INFO -- : Running discovery modules...
I, [2017-07-24T23:20:47.557092 #833] INFO -- : Running discovery module: GetContainers
I, [2017-07-24T23:20:47.559727 #833] INFO -- : Running discovery module: GetDockerInfo
I, [2017-07-24T23:20:47.591014 #833] INFO -- : Running discovery module: GetContainersRunning
I, [2017-07-24T23:20:47.595073 #833] INFO -- : Running discovery module: GetDockerVersion
I, [2017-07-24T23:20:47.596782 #833] INFO -- : Running discovery module: GetImages
I, [2017-07-24T23:20:47.622201 #833] INFO -- : Loading audit modules...
I, [2017-07-24T23:20:47.622613 #833] INFO -- : Loading audit module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/audit/container-number-process.rb
I, [2017-07-24T23:20:47.623316 #833] INFO -- : Loading audit module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/audit/docker-limits.rb
I, [2017-07-24T23:20:47.623786 #833] INFO -- : Loading audit module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/audit/docker-storage-driver-aufs.rb
I, [2017-07-24T23:20:47.624261 #833] INFO -- : Loading audit module: /var/lib/gems/2.3.0/gems/dockscan-0.1.2/audit/docker-storage-driver-diff
```

Homepage: github.com/kost/dockscan

License: Open Source

Use Cases: Compliance & audit

A simple ruby script that analyzes the Docker installation and running containers, both for local and remote hosts.

It's easy to install and run with just one command and can generate HTML report files. Dockscan reports configured resource limits, containers spawning too many processes or with a high number of modified files, or if your Docker host is allowing containers to directly forward traffic to the host gateway, to name a few checks.

HashiCorp Vault



Hashicorp's Vault is an advanced suite for managing secrets: Passwords, SSL/TLS certificates, API keys, access tokens, SSH credentials, etc. It supports time-based secret leases, fine-grained secret access, on-the-fly generation of new secrets, key rolling (renewing keys without losing access to secrets generated using the old one) and much more.

Vaults keeps a detailed audit log to keep track of all the secrets and the access and manipulations performed by each user/entity, so operators can easily trace any suspicious interaction.

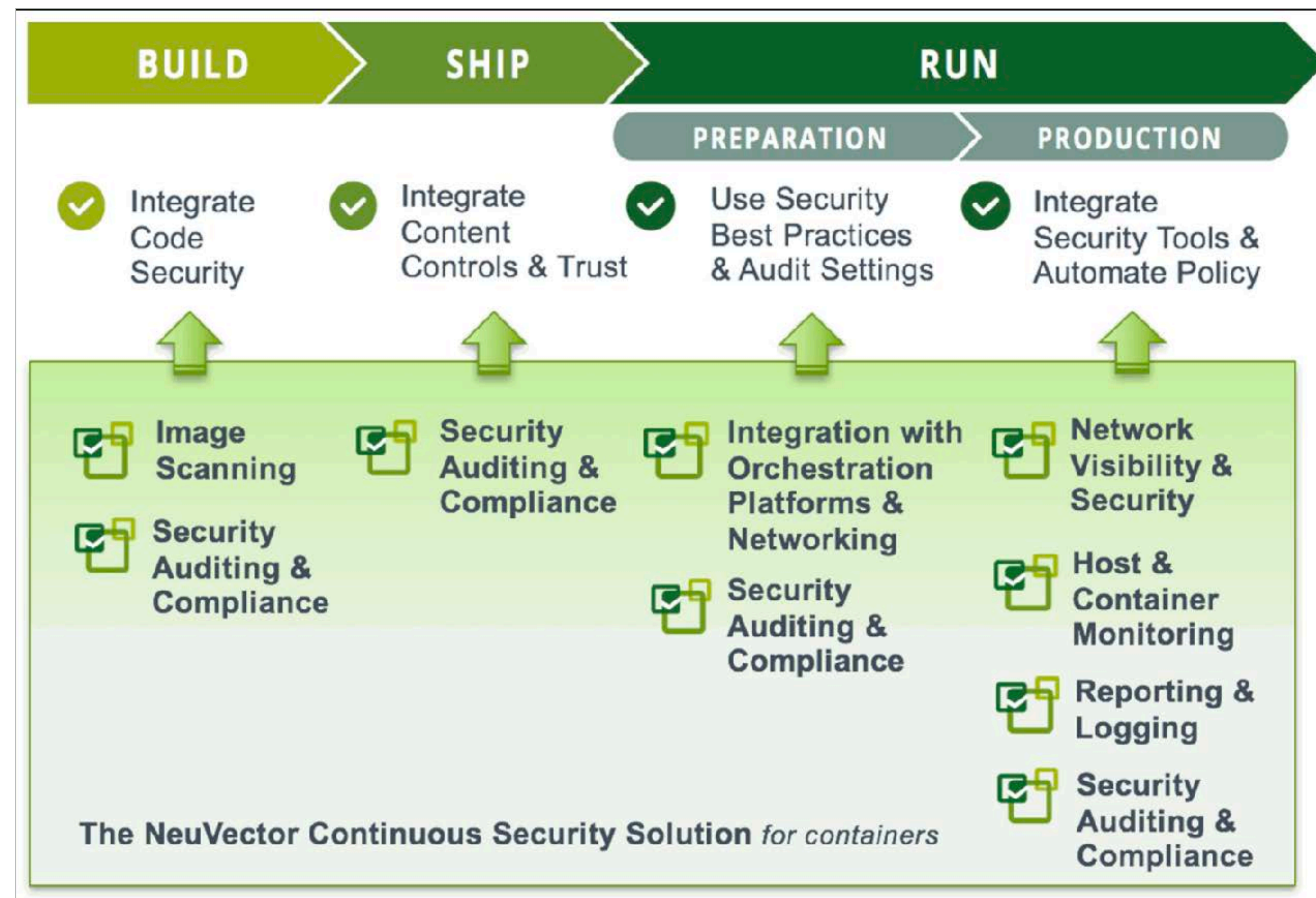
Docker context: The secure distribution and traceability of secrets is a core concern in the new microservices and containerized environments, where software entities are constantly spawned and deleted. Vault itself can be deployed as a Docker container.

Homepage: vaultproject.io

License: Free with enterprise version

Use Cases: Secure container-aware credentials storage, and trust management

NeuVector



NeuVector focuses on real-time security protection at runtime. Automatically discovers behavior of applications, containers, and services, detects security escalations and other related threats in a similar fashion to other Linux IDS. NeuVector privileged 'enforcer' containers are deployed on each physical host, with full access to the local Docker daemon, apart from that, the internal technology used by NeuVector is not thoroughly detailed in the publicly accessible documentation.

NeuVector aims to be a non-intrusive, plug&play security suite, performing automatic discovery of running containers and their default behavior to assist and counsel the operators in the design of their infrastructure security profiles.

Homepage: neuvector.com

License: Commercial

Use Cases: Runtime protection, compliance & audit

Notary

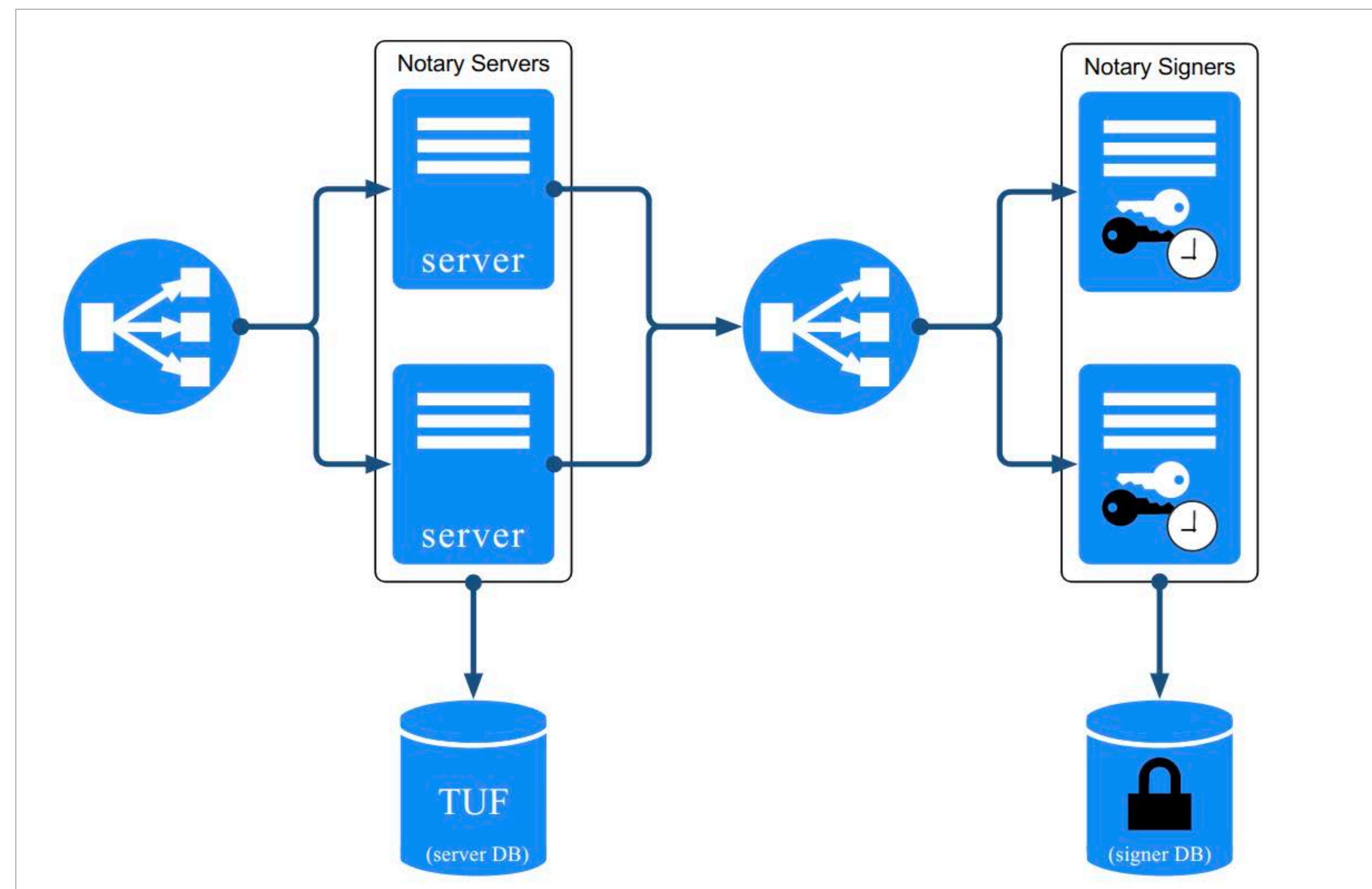


Image forgery and tampering is one major security concern for Docker-based deployments. Notary is a tool for publishing and managing trusted collections of content. You can approve trusted published and create signed collections, in a similar way to the software repository management tools present in modern Linux systems, but for Docker images.

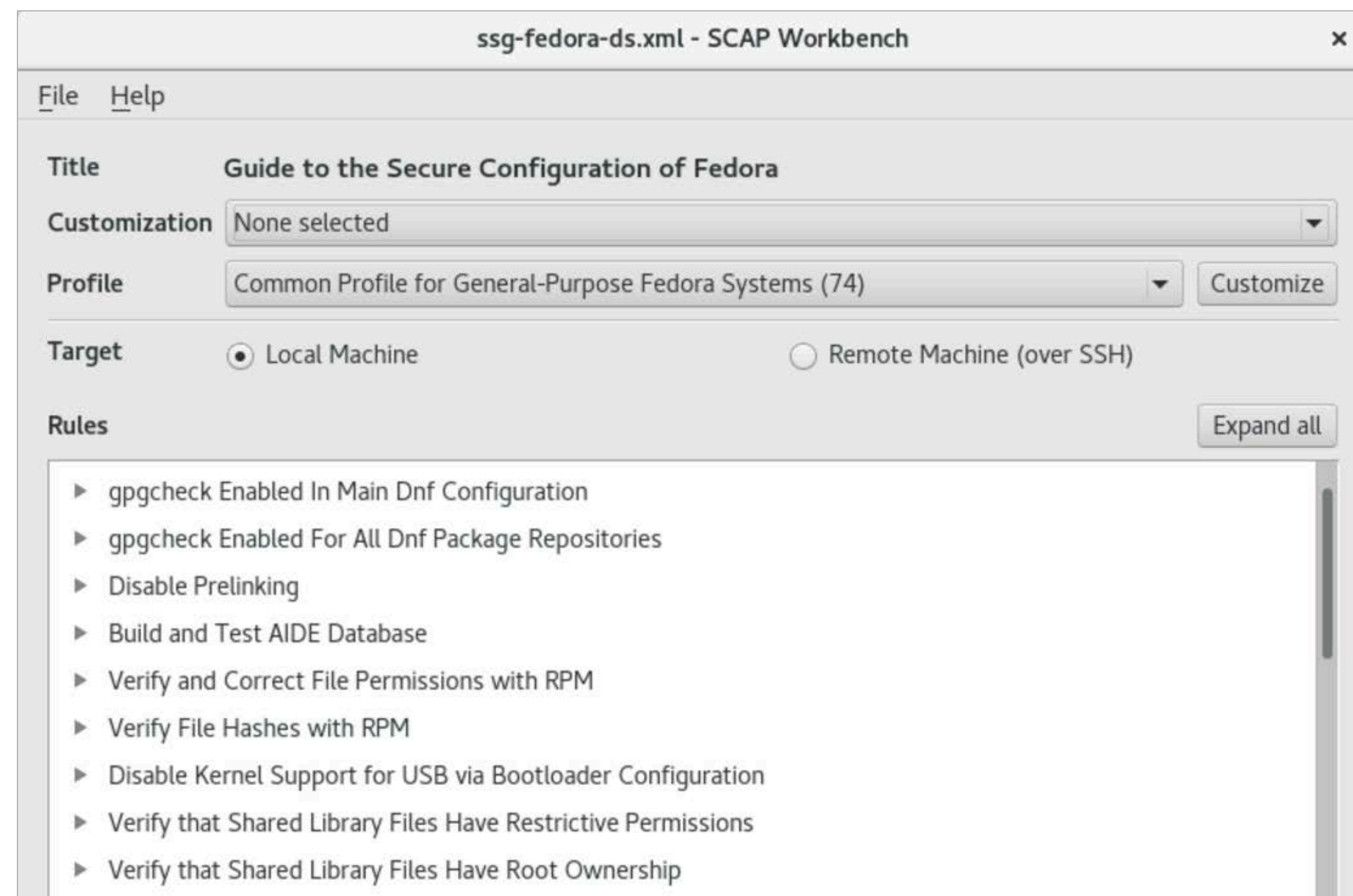
Some of Notary goals include guaranteeing image freshness (most up to date content, to avoid known vulnerabilities), trust delegation between users or trusted distribution over untrusted mirrors or transport channels.

Homepage: github.com/docker/notary

License: Open Source

Use Cases: Trusted image repository, trust management, and verifiability

OpenSCAP



Homepage: open-scap.org

License: Open Source

Use Cases: Compliance & audit, certification

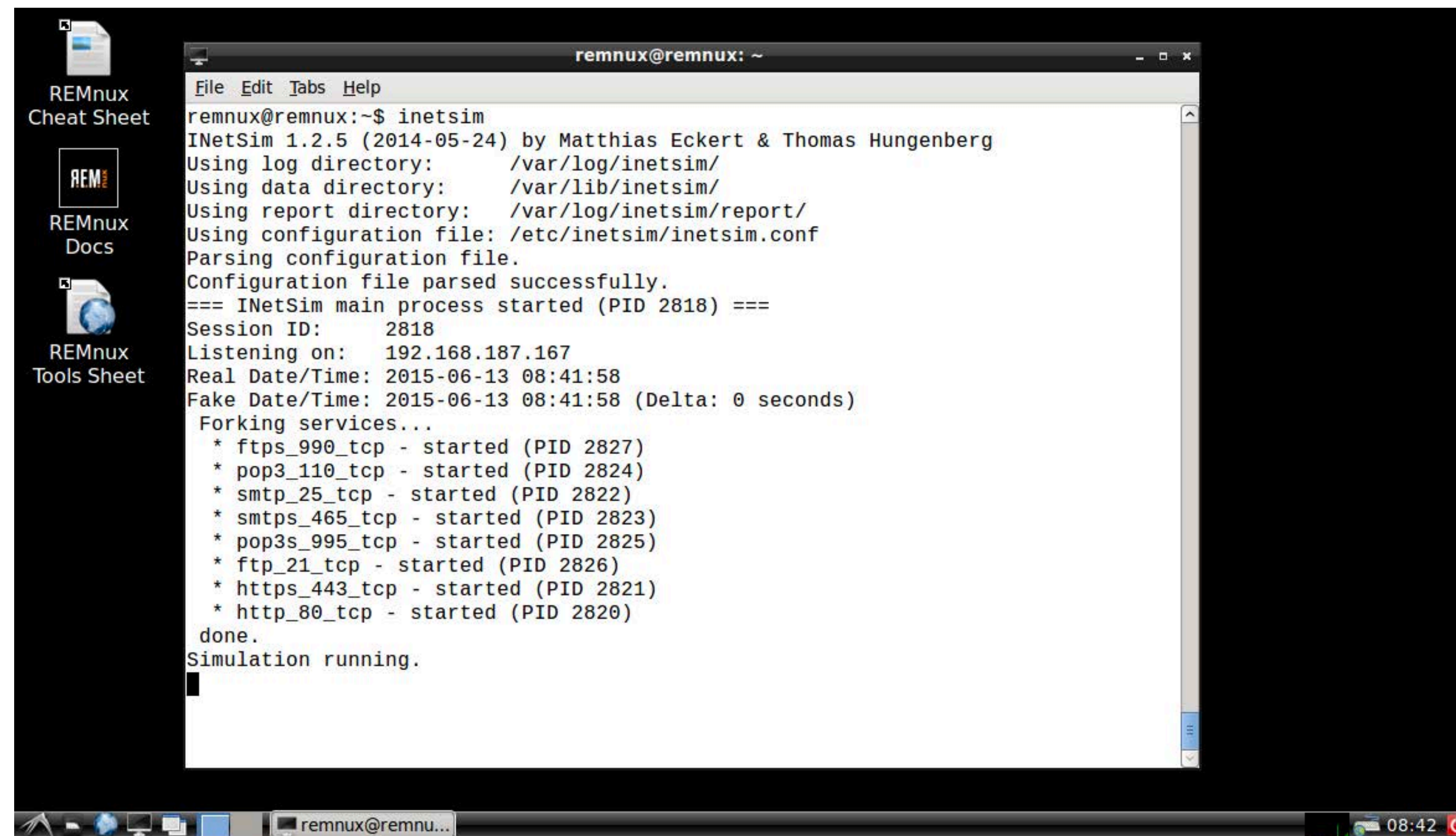
OpenSCAP provides a suite of automated audit tools to examine the configuration and known vulnerabilities in your software, following the NIST-certified Security Content Automation Protocol (SCAP).

You can create your own custom assertions and rules and routinely check that any software deployed in your organization strictly abides.

These set of tools is not only focused on the security itself, but also on providing the formal tests and reports that you may need to meet an official security standard.

Docker context: The OpenSCAP suite provides a Docker-specific tool [oscap-docker](#) to audit your images, assessing both running containers and cold images.

REMnux



```
remnux@remnux: ~  
File Edit Tabs Help  
remnux@remnux:~$ inetsim  
INetSim 1.2.5 (2014-05-24) by Matthias Eckert & Thomas Hungenberg  
Using log directory: /var/log/inetsim/  
Using data directory: /var/lib/inetsim/  
Using report directory: /var/log/inetsim/report/  
Using configuration file: /etc/inetsim/inetsim.conf  
Parsing configuration file.  
Configuration file parsed successfully.  
=== INetSim main process started (PID 2818) ===  
Session ID: 2818  
Listening on: 192.168.187.167  
Real Date/Time: 2015-06-13 08:41:58  
Fake Date/Time: 2015-06-13 08:41:58 (Delta: 0 seconds)  
Forking services...  
* ftps_990_tcp - started (PID 2827)  
* pop3_110_tcp - started (PID 2824)  
* smtp_25_tcp - started (PID 2822)  
* smtps_465_tcp - started (PID 2823)  
* pop3s_995_tcp - started (PID 2825)  
* ftp_21_tcp - started (PID 2826)  
* https_443_tcp - started (PID 2821)  
* http_80_tcp - started (PID 2820)  
done.  
Simulation running.
```

Homepage: remnux.org

License: Open Source

Use Cases: Forensics

A security oriented distribution based on Ubuntu. REMnux is a free Linux toolkit for assisting malware analysts with reverse-engineering malicious software, commonly known as forensics. As you can guess, this system bundles a vast amount of pre installed analysis and security tools: Wireshark, ClamAV, tcpextract, Rhino debugger, Sysdig, vivisect... just to name a few.

REMnux aims to be swiss knife that you carry around in a usb memory in case you suspect any of your systems have been compromised.

Docker context: The REMnux project conveniently provides several of its integrated security tools as Docker containers, so you can instantly launch difficult-to-install security applications when you most need them.

SELinux

```
package selinux
import (
    "fmt"
    "k8s.io/kubernetes/pkg/api"
    "k8s.io/kubernetes/pkg/apis/extensions"
    "k8s.io/kubernetes/pkg/util/validation/field"
)
type mustRunAs struct {
    opts *extensions.SELinuxStrategyOptions
}
var _ SELinuxStrategy = &mustRunAs{}
func NewMustRunAs(options *extensions.SELinuxStrategyOptions) (SELinuxStrategy, error) {
    if options == nil {
        return nil, fmt.Errorf("MustRunAs requires SELinuxContextStrategyOptions")
    }
    if options.SELinuxOptions == nil {
        return nil, fmt.Errorf("MustRunAs requires SELinuxOptions")
    }
    return &mustRunAs{
        opts: options,
    }
}
```

Homepage: selinuxproject.org

License: Open Source

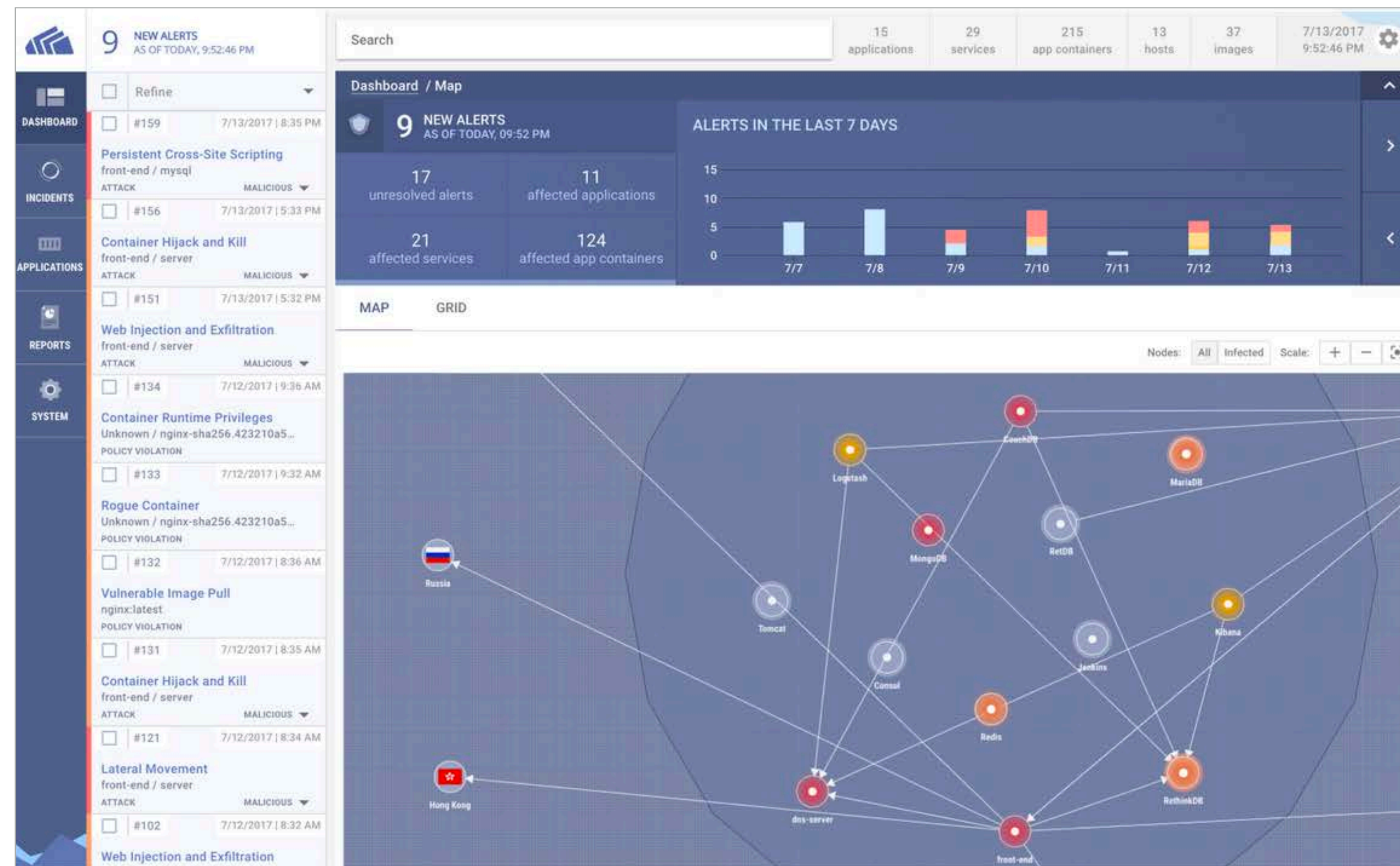
Use Cases: Runtime protection,
Mandatory Access Control (MAC)

Security-Enhanced Linux (SELinux) is a Linux kernel security module. It is often compared with [AppArmor](#), and it's also a Mandatory Access Control system. SELinux provides security capabilities from mandatory access controls to mandatory integrity controls, role-based access control (RBAC) and type enforcement architecture.

SELinux has a reputation of being particularly complex but powerful, fine-grained and flexible.

Docker context: Similarly to AppArmor, SELinux offers an extra layer of access policies and isolation between the host and the containerized apps.

StackRox



StackRox feature proposal revolves around the concepts of "Adaptive security" and auto discovery of components and behaviours. Highly focused on machine learning, automatic even correlation and dynamic pattern recognition, StackRox aims to provide fast-response, minimum effort security that will evolve hand on hand with your platform. Apart from the machine learning component, StackRox provides the usual features of commercial security platforms like cold image scanning or default security profiles ala SELinux.

Homepage: stackrox.com

License: Commercial

Use Cases: Runtime protection, machine learning, pre-production analysis

Seccomp

```
{
  "defaultAction": "SCMP_ACT_ERRNO",
  "architectures": [
    "SCMP_ARCH_X86_64",
    "SCMP_ARCH_X86",
    "SCMP_ARCH_X32"
  ],
  "syscalls": [
    {
      "name": "accept",
      "action": "SCMP_ACT_ALLOW",
      "args": []
    },
    {
      "name": "accept4",
      "action": "SCMP_ACT_ALLOW",
      "args": []
    },
    {
      "name": "access",
      "action": "SCMP_ACT_ALLOW",
      "args": []
    },
    {
      "name": "alarm",
      "action": "SCMP_ACT_ALLOW",
      "args": []
    }
  ]
}
```

Homepage: kernel.org

License: Open Source

Use Cases: Runtime protection,
Mandatory Access Control (MAC)

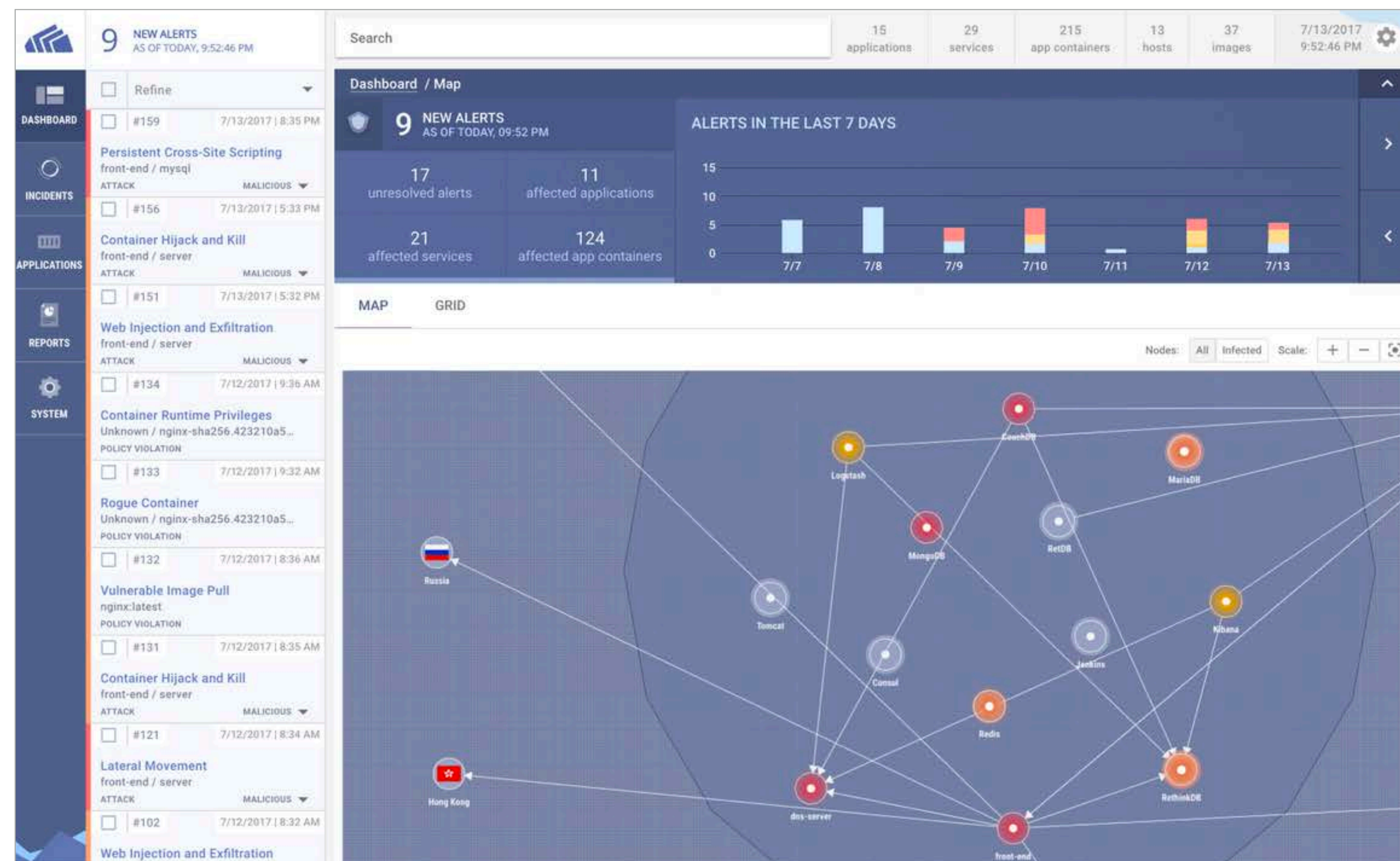
Seccomp is not so much a tool but rather a sandboxing facility in the Linux kernel. You can think of it as an iptables rules-based firewall but for system calls. It uses Berkeley Packet Filter (BPF) rules to filter syscalls and control how they are handled.

With Seccomp you can selectively choose which syscalls are forbidden/allowed to each container. For example, you can forbid file-permissions manipulations inside your container.

You may have noticed the similarities with Falco, both are closely related to the Linux Syscall API. This article compares these two (with AppArmor and SELinux) solutions. TL;DR: Unlike the others, Falco integrates rich high level container specific context to build rules.

Docker context: Docker has used Seccomp since version 1.10 of the Docker Engine, Docker has its own JSON-based DSL that allows you to define profiles that will be compiled to seccomp filters.

StackRox



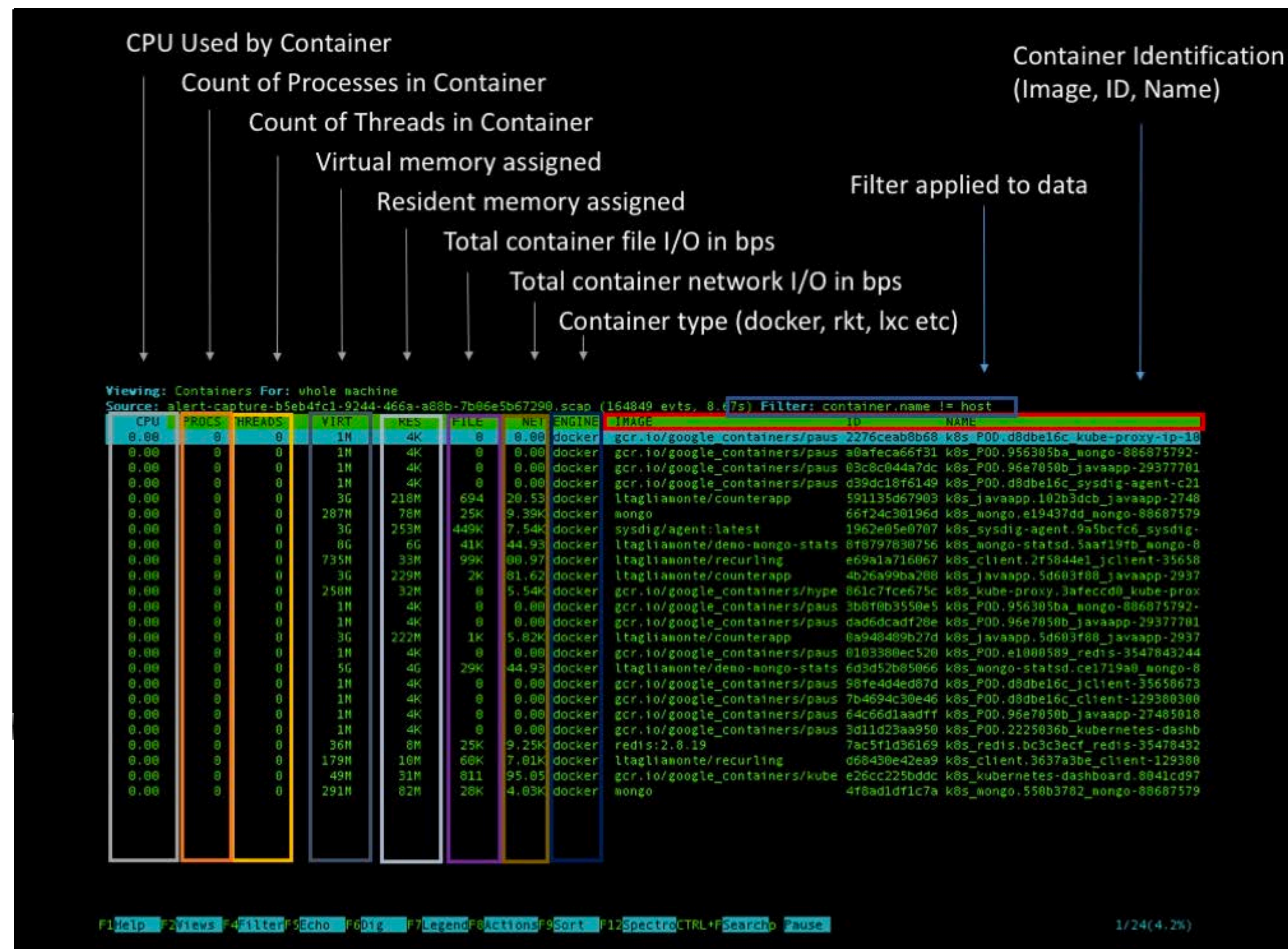
StackRox feature proposal revolves around the concepts of "Adaptive security" and auto discovery of components and behaviours. Highly focused on machine learning, automatic even correlation and dynamic pattern recognition, StackRox aims to provide fast-response, minimum effort security that will evolve hand on hand with your platform. Apart from the machine learning component, StackRox provides the usual features of commercial security platforms like cold image scanning or default security profiles ala SELinux.

Homepage: stackrox.com

License: Commercial

Use Cases: Runtime protection, machine learning, pre-production analysis

Sysdig



Sysdig is a full-system exploration, troubleshooting and debugging tool for Linux systems. It records all system calls made by any process, allowing system administrators to find bugs in the operating system or any processes running on it.

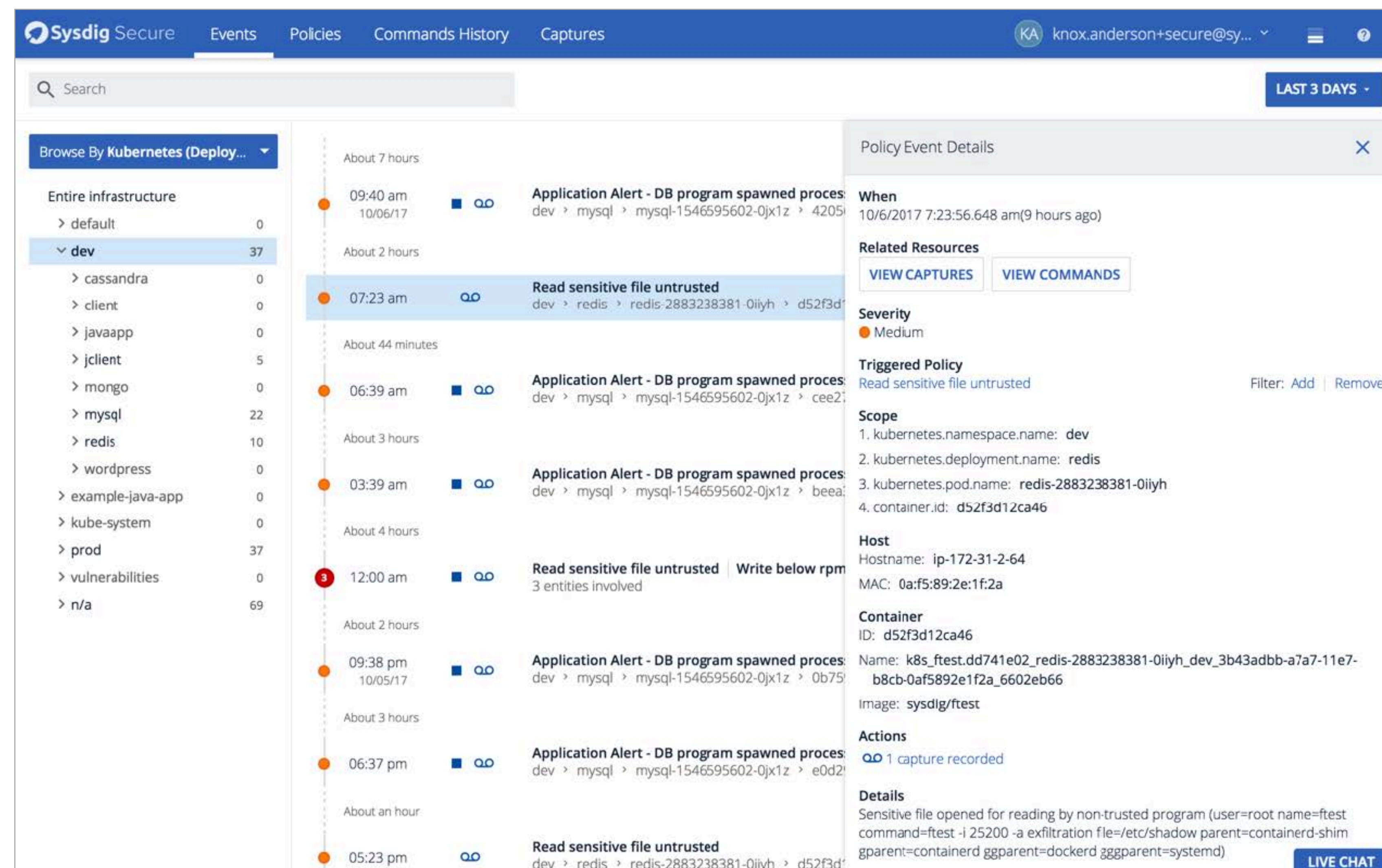
Sysdig has a command line interface with a syntax similar to tcpdump and a ncurses interface to visually navigate and filter through the events, in a similar fashion to htop or Wireshark. The system call capture files allow you to perform forensics on your containers even if they are long gone.

Homepage: sysdig.org

License: Open source, commercial products built on top of the free technology

Use Cases: Anomalous behaviour debugging, forensics

Sysdig Secure



Sysdig Secure is a powerful run-time security and forensics solution for your containers and microservices. Secure is part of the Sysdig Container Intelligence Platform, and as the rest of the family comes out-of-the-box with deep container visibility and container orchestrator tools integration, including Kubernetes, Docker, AWS ECS, and Mesos.

Sysdig Secure protects your entire infrastructure: containers & hosts as well as the logical services that run on top of them. Sysdig Secure also provides full stack forensics capabilities for pre and post attack investigation.

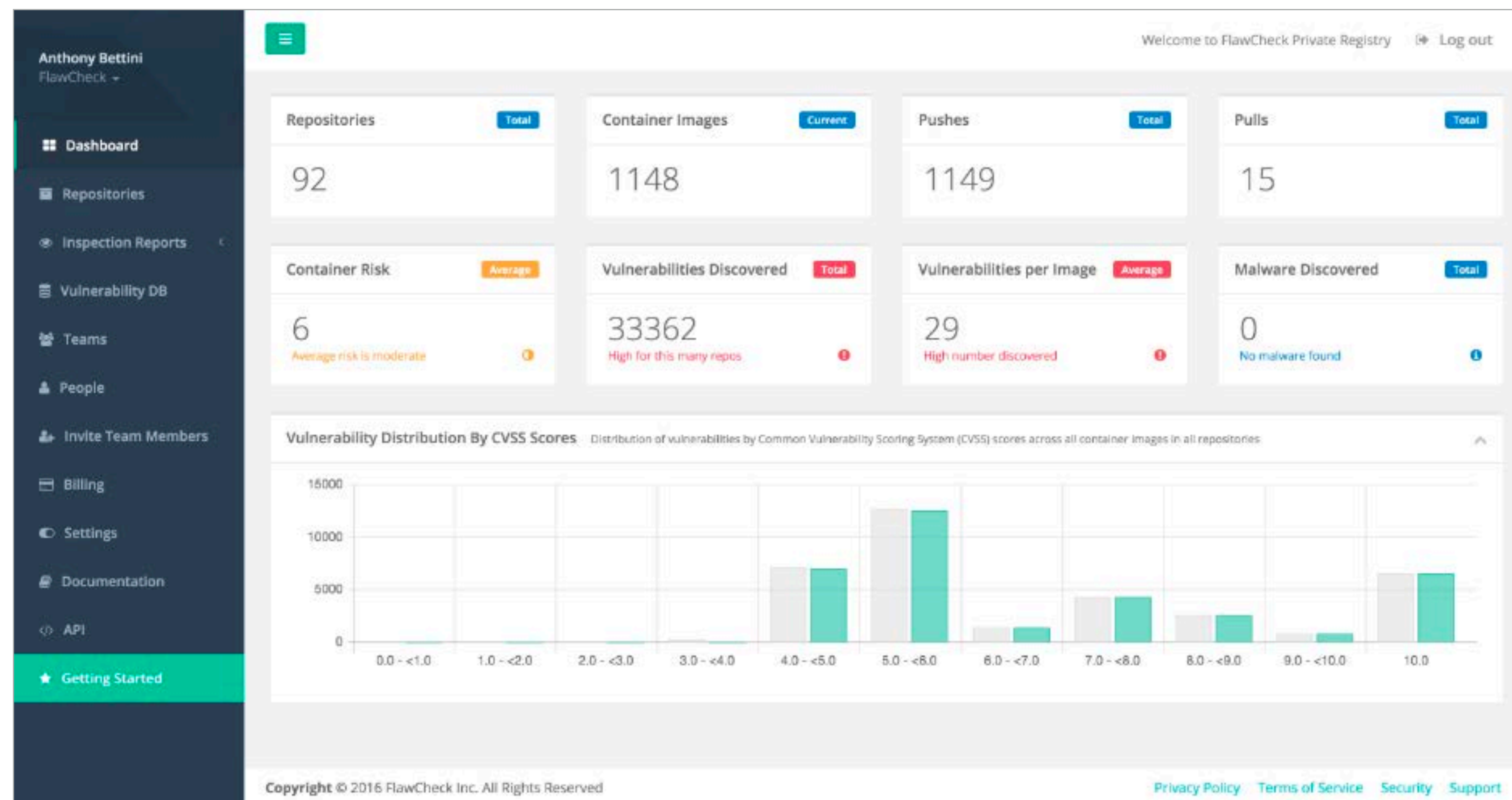
Sysdig provides full performance monitoring and troubleshooting for your environment. A single instrumentation both for monitoring and security with no added overhead.

Website: sysdig.com/product/secure

License: commercial

Use cases: runtime security, forensics and audit, hybrid environments (containers and traditional deployment), performance monitoring & troubleshooting, available both as SaaS and on-prem.

Tenable Flawcheck



Tenable, the company perhaps best known for Nessus, the security scanner, acquired Flawcheck, a specific container-focused security solution.

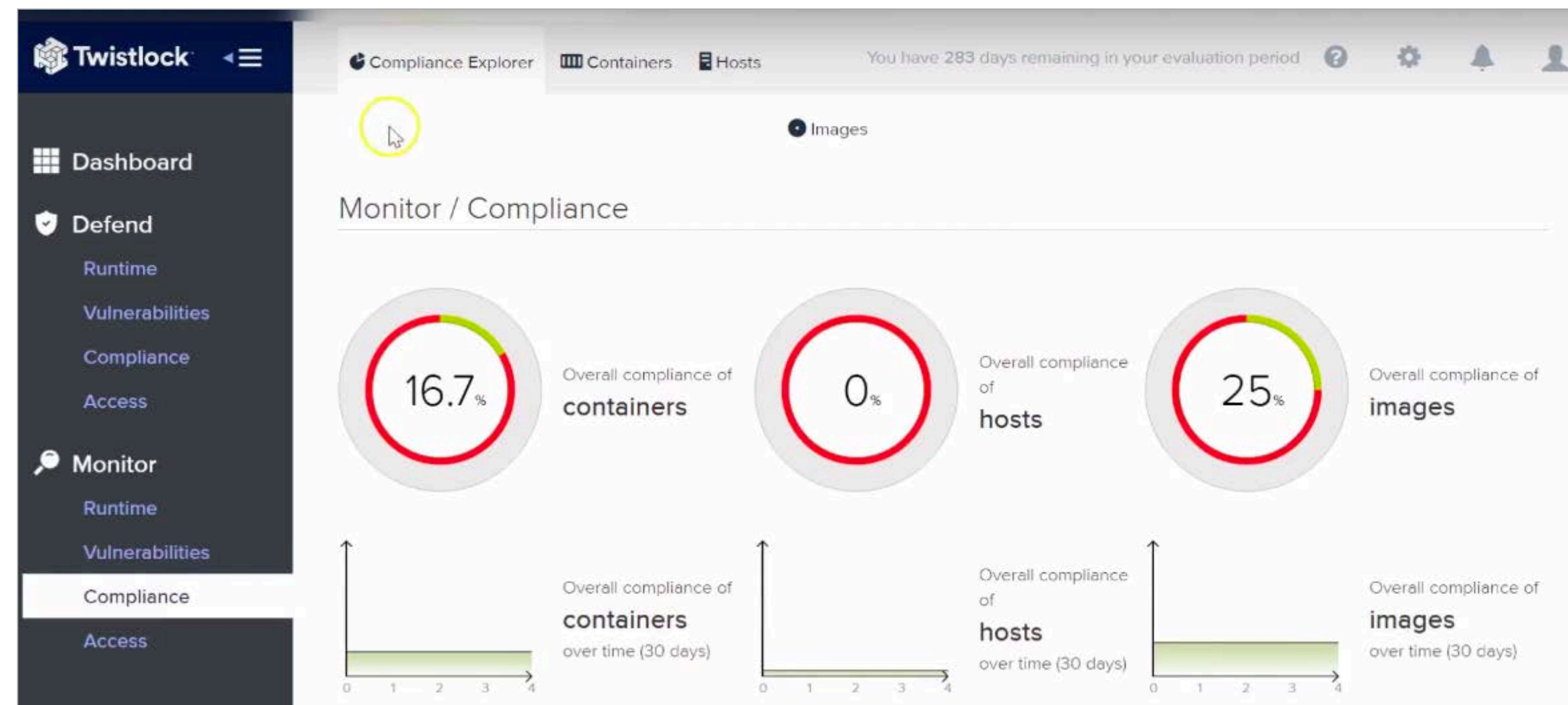
FlawCheck, like other commercial tools in this list, stores container images and scans them as they're built, before they can reach production. FlawCheck leverages Tenable/Nessus know-how and database of vulnerabilities, malware and intrusion vectors and adapts it to containerized and agile CI/CD environments.

Homepage: tenable.com/flawcheck

License: Commercial

Use Cases: Pre-production analysis, vulnerability newsfeed

Twistlock



A commercial security suite built to support containerized environments: vulnerability management, access control, analytics and forensics to security standards compliance.

Twistlock integrates with your continuous integration / continuous delivery pipeline, providing native plugins for popular tools like Jenkins or TeamCity and callable webhooks, so you can trigger the indexing and scanning process for every build and testing environment.

Homepage: twistlock.com

License: Commercial

Use Cases: Pre-production analysis, runtime protection, compliance & audit, etc.

Conclusion

We hope you find this Docker security tools list useful. If you have suggestions or additional tools we should add, feel free to ping us at [@sysdig](#) or reach us on the [Sysdig community Slack group](#).