无标题-4

|  |  |  |  |
| --- | --- | --- | --- |
| 业界镜像安全的发展趋势分析 | | | |
| ■ 文档编号 | 请输入文档编号 | ■ 密级 | 内部使用 |
| ■ 版本编号 |  | ■ 日期 |  |



|  |
| --- |
| © 2017 绿盟科技 |

|  |
| --- |
| ■ 版权声明 |
| 本文中出现的任何文字叙述、文档格式、插图、照片、方法、过程等内容，除另有特别注明，版权均属**绿盟科技**所有，受到有关产权及版权法保护。任何个人、机构未经**绿盟科技**的书面授权许可，不得以任何方式复制或引用本文的任何片断。 |

|  |  |  |  |
| --- | --- | --- | --- |
| ■ 版本变更记录 | | | |
| 时间 | 版本 | 说明 | 修改人 |
| 2017.11 | v1 | 初稿 | 江国龙、范楷朋 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| ■ 适用性声明 |
| 本模板用于撰写绿盟科技内外各种正式文件，包括技术手册、标书、白皮书、会议通知、公司制度等文档使用。 |

|  |
| --- |
| 目录 |
| [一. 简介 4](#_Toc497987759)  [1.1 关键发现 4](#_Toc497987760)  [1.2 研究方法 4](#_Toc497987761)  [二. 常见物联网设备的暴露情况 4](#_Toc497987762)  [2.1 引言 4](#_Toc497987763)  [2.2 视频监控设备 4](#_Toc497987764)  [2.2.1 总体情况 4](#_Toc497987765)  [2.2.2 特定厂商 4](#_Toc497987766)  [2.3 物联网设备分析小结 4](#_Toc497987767)  [三. 物联网操作系统的暴露情况 4](#_Toc497987768)  [3.1 引言 5](#_Toc497987769)  [3.2 Openwrt 5](#_Toc497987770)  [3.3 操作系统分析小结 5](#_Toc497987771)  [四. 总结 5](#_Toc497987772)  [参考资料： 5](#_Toc497987773) |

|  |
| --- |
| 表格索引 |
| **未找到目录项。** |

|  |
| --- |
| 插图索引 |
| **未找到目录项。** |

1. 简介

微服务架构的兴起，容器化部署已经成为时下最流行的生产方式，越来越多的公司将应用部署在基于容器的架构上。自然的，随着容器的广泛使用，容器的安全性就成为了业界关注的焦点，已经有几家容器安全厂商如雨后春笋般相继成立，如：CoreOSClair、AquaSecurity、Twistlock、Anchore等等。

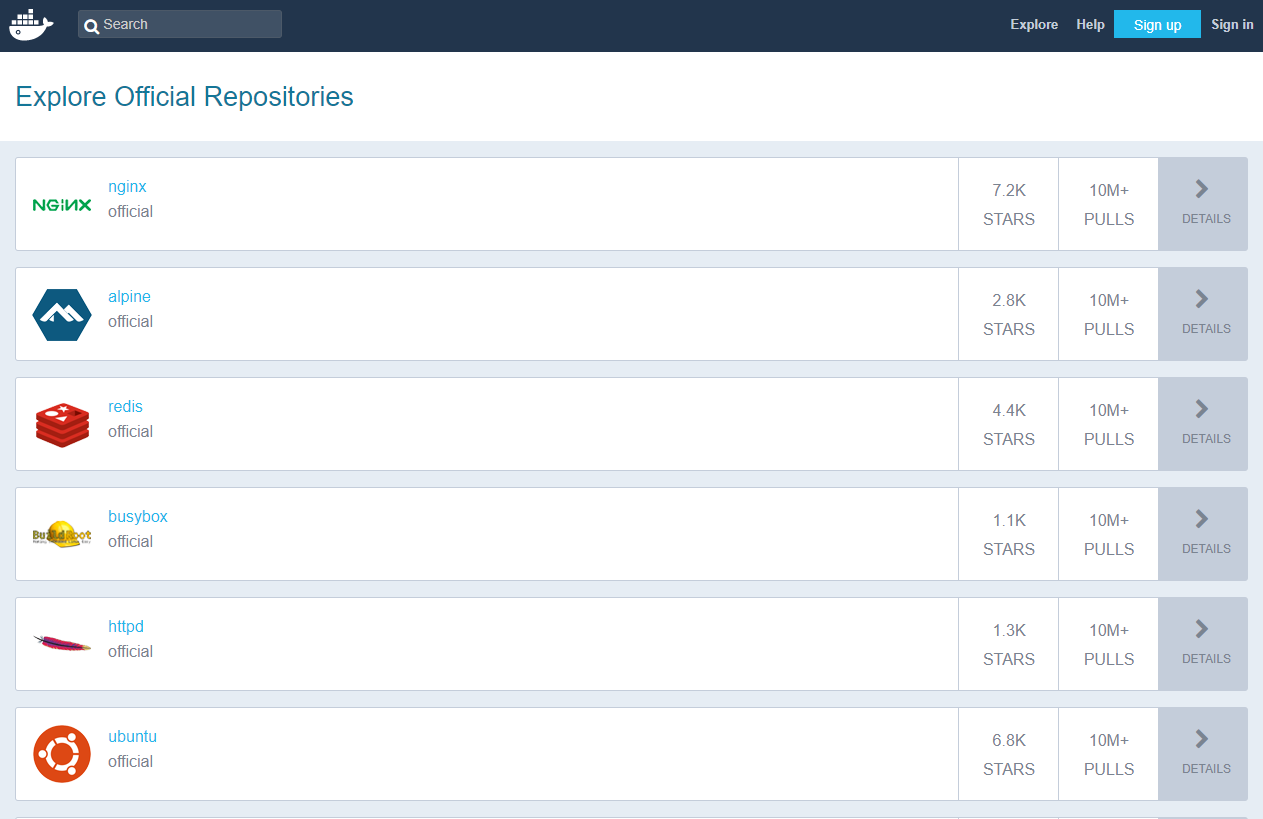
容器是基于镜像构建的，如果镜像本身就是一个恶意镜像或是一个存在漏洞的镜像，那么基于它搭建的容器自然就是不安全的了，故镜像安全直接决定了容器安全。

接下来，我们就来谈谈业界镜像安全的发展......

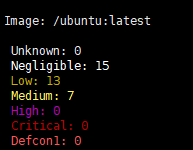
1. 业界镜像的安全情况
   1. 引言

Docker Hub是全球最大的镜像仓库，如ubuntu、centos、apache这些热门镜像每天的下载量数以万计，倘若这些镜像存在漏洞，那么当广大用户使用部署时，受到的安全威胁程度将是巨大的。

所以，接下来，我们就来看看仓库中的几大热门镜像安全性如何吧

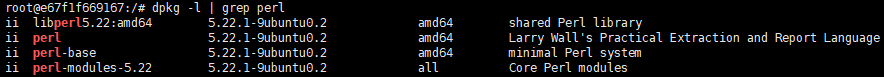


* 1. Ubuntu / 16.04.3

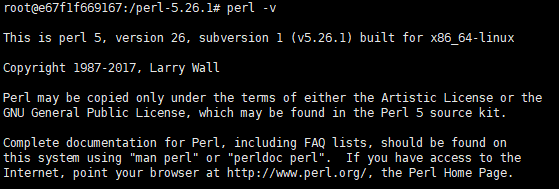




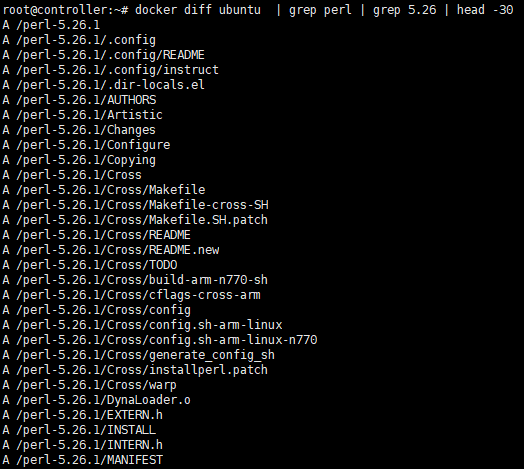
Ubuntu:16.04作为最大众的操作系统镜像，其安全性还是挺不错的。虽然有35个漏洞，但大部分都是可忽略和低危漏洞，只有7个中危漏洞，分别为：apparmor(1)、perl(2)、db5.3(1)、systemd(1)、glibc(2)



下载http://www.cpan.org/src/5.0/perl-5.26.1.tar.gz并安装



Perl已升级到5.26

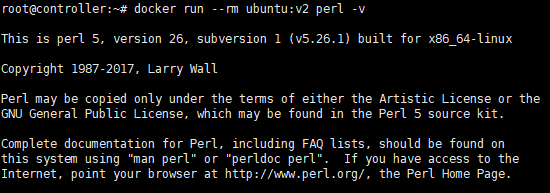


可以看到对ubuntu镜像的改动

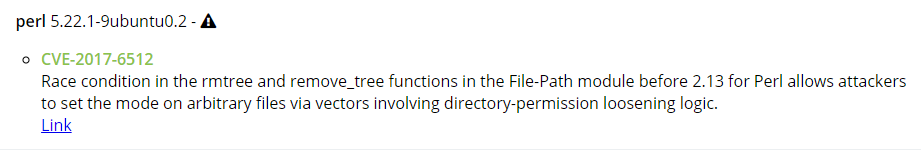
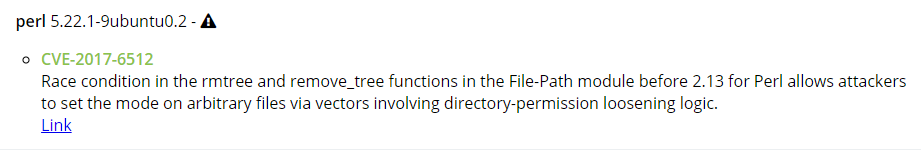
将更新了perl后的ubuntu打包成新的镜像ubuntu:v2



确认ubuntu:v2镜像中的perl确实是最新版本

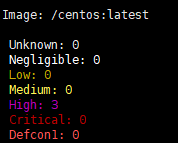


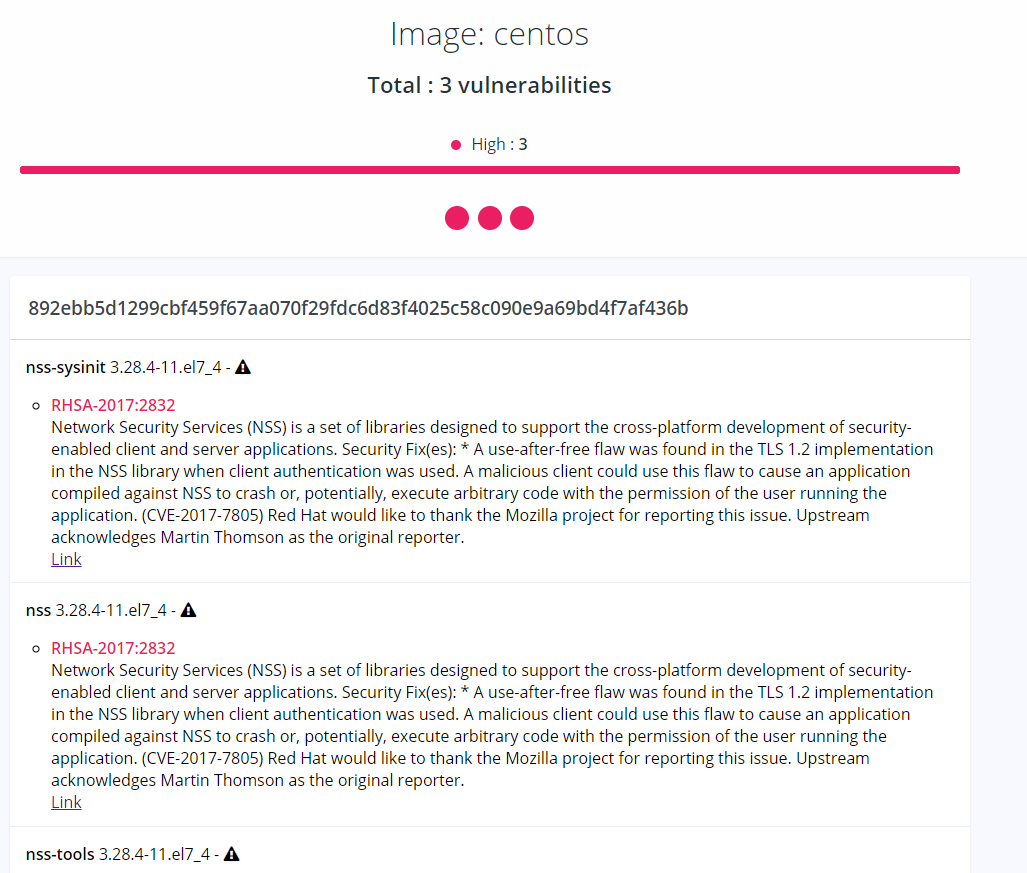
再次使用clair进行扫描，发现还是会报告perl5.22版本存在漏洞



这种误报的原因就是：用户上传镜像至Clair进行扫描时，是将镜像中的每个layer依次上传分别扫描的，我们虽然手动更新了perl，但所有更新的操作都被封装在了新的一个layer中，当Clair扫描到之前装有perl:5.22这个漏洞软件的layer的时候，就已经发出警报了，所以Clair在用户手动修补漏洞的情况下会产生误报，这种情况在类似的镜像扫描器（比如anchore）中也会出现

* 1. Centos / 7

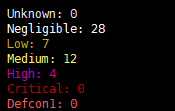


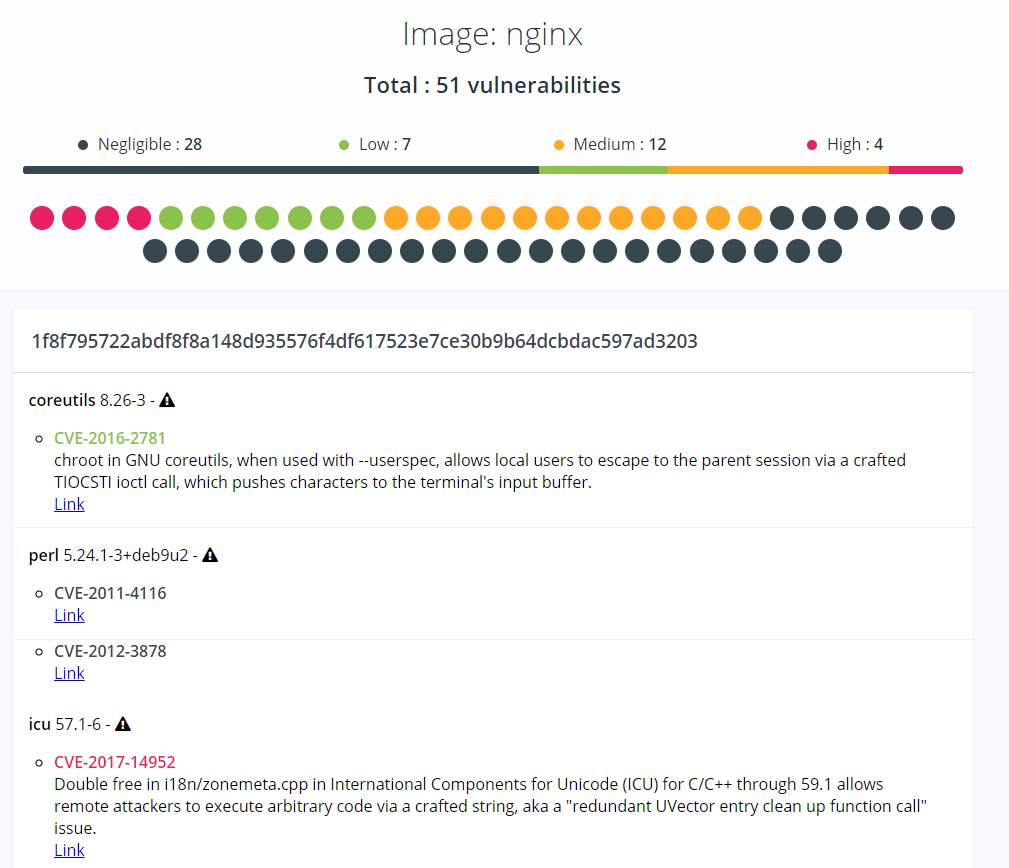


Centos:7镜像虽然漏洞不多，但是却有3个高危漏洞，分别是：

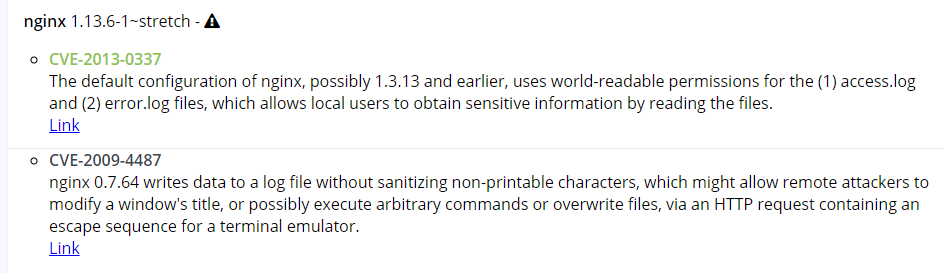
nss-sysinit(1)、nss(1)、nss-tools(1)

* 1. Nginx / 1.13.6

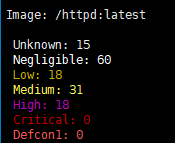


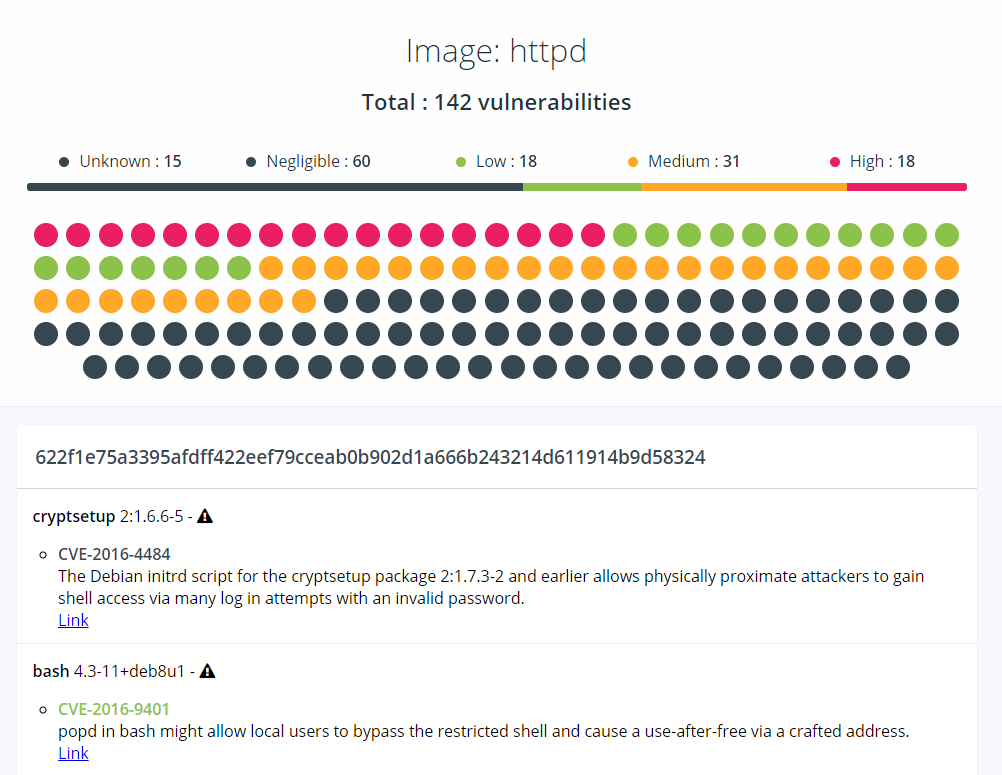


Nginx:1.13虽然有多达51个漏洞(包括4个高危漏洞)，但基本上是它的父layer(debian9)的漏洞，nginx自身的漏洞只有这两个低危漏洞

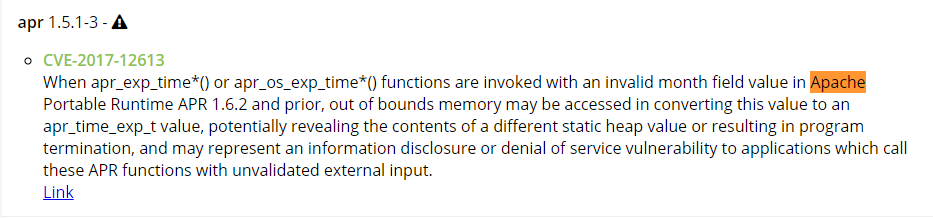


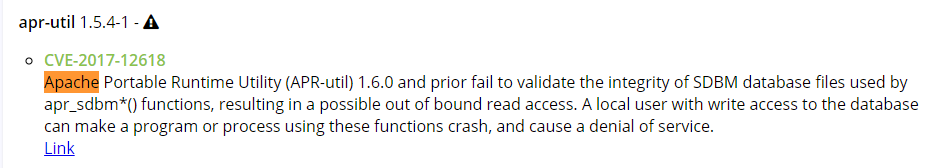
* 1. Apache / 2.4.29





同样的原理，apache:2.4.29因为搭建在debian8的layer之上，所以被扫描出的漏洞更多，但是apache自身的漏洞只有两个





* 1. 镜像安全情况小结

综上所述，ubuntu、centos等最新版操作系统存在的漏洞不多，其中的高危更是少之又少，所以用户下载最新版OS镜像时，受到的安全威胁不大。相比之下，最新版的热门应用(nginx、apache等)搭建在旧版本OS镜像(debian9、debian8)之上，所爆出的漏洞就很多了，所以大家使用APP镜像前先检查其基于的OS的版本，若版本过低，则建议大家自行在最新版OS上搭建新的APP镜像

1. 业界镜像安全的解决方案
   1. 引言

上一节我们分析了Docker Hub上的镜像安全情况，发现许多热门镜像都存在漏洞，那么我们该如何在使用之前发现这些漏洞呢？还有种情况，Docker Hub允许用户自己上传镜像，倘若一个黑客将自己制作的恶意镜像上传到Docker Hub上，用户不知情的情况下下载并使用了，这种问题又如何避免呢。

好的，接下来我们就来介绍介绍基于漏洞镜像和恶意镜像的安全扫描解决方案

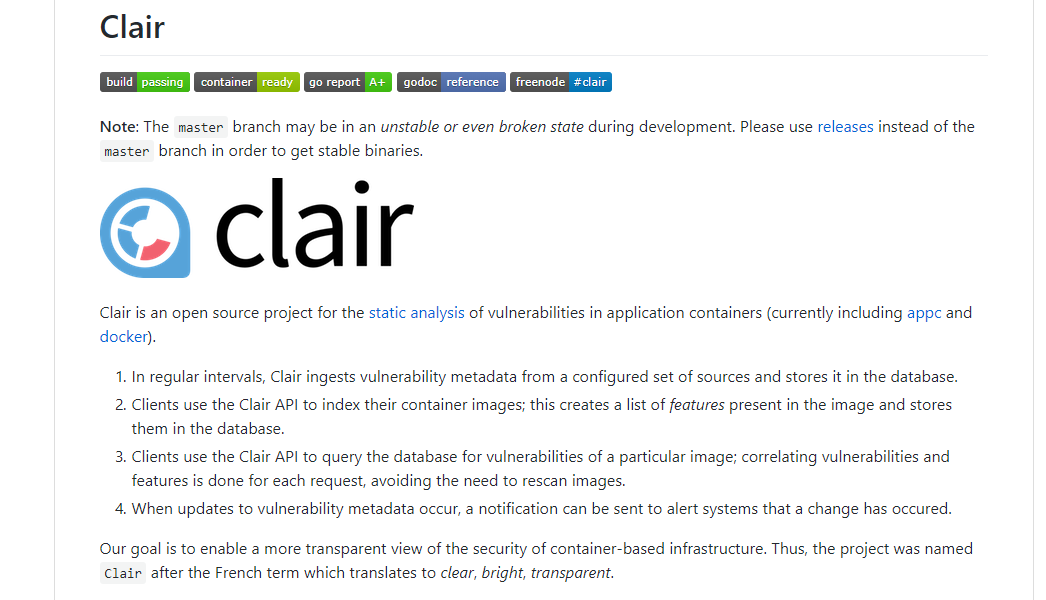
* 1. 漏洞镜像安全扫描

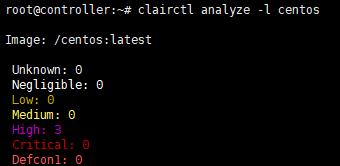
每天各大安全网站都会爆出最新的CVE漏洞，自然的我们的镜像就会立刻变成一个漏洞镜像，那么我们如何快速地扫描出镜像是否存在漏洞呢？这就必须提起CoreOS开发的一个镜像扫描利器:Clair

* + 1. Clair

Clair的目标是能够从一个更加透明的维度去看待基于容器化的基础框架的安全性、是由coreos所推出的这样一款针对容器镜像的安全扫描的工具。

Clair主要模块分为Detector、Fetcher、Notifier和Webhook，Clair首先对镜像进行特征的提取，然后再将这些特征匹配CVE漏洞库，若发现漏洞则进行提示及修补，其功能侧重于扫描容器中的OS及APP的CVE漏洞





Clair通过对镜像中的特征进行提取，然后比对其强大的漏洞库，然后反馈给用户扫描信息，来实现对镜像的安全评估。

Clair的基本原理如下：

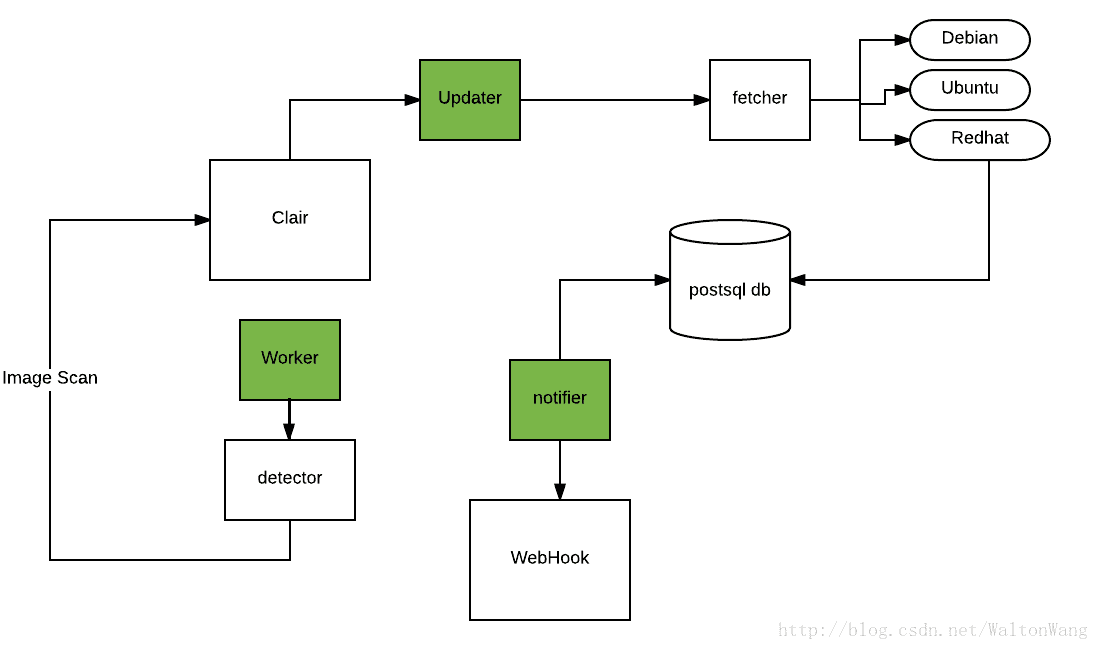
1. 用户调用Clair的Restful API上传image的每一个layer，对应每一个layer都会起一个worker对其操作

2. 每一个worker都会启动一个Detector来对layer进行扫描

3. Detector的先对上传上来的layer压缩包进行解压，然后解析出OS的类型及版本，最后解析出安装应用的名及版本

4. Updater启动一个Fetcher，Fetcher根据layer的OS查找对应的CVE漏洞库，查看当前OS版本是否存在CVE漏洞，以及扫描当前OS下的应用是否也存在漏洞

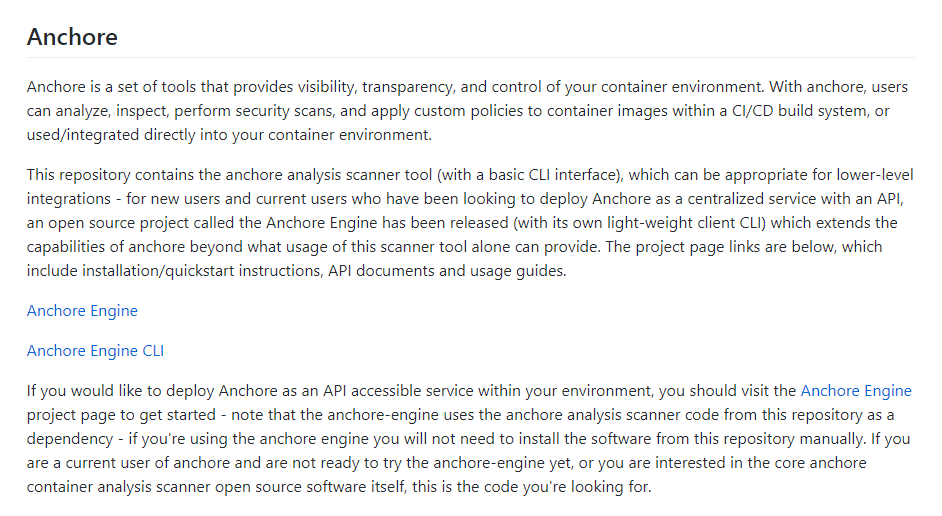
5. 这是主流程，还有一个异步的流程，Notifier+WebHook，WebHook一直在各大CVE网站爬取最新的漏洞，当新的CVE漏洞爆发时，及时通知Notifier，然后将漏洞信息更新到Postgresql数据库



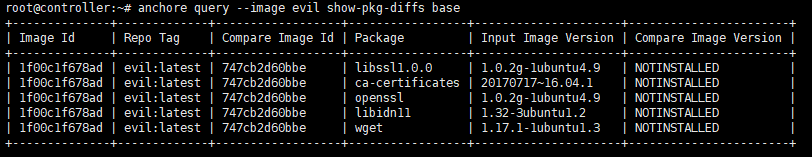
* 1. 恶意镜像

Clair能扫描出一个镜像中的所有CVE漏洞，但现在有一种情况，黑客使用最新版无漏洞的OS镜像，然后在其之上安装后门木马，或执行恶意命令，这样Clair就不能检测其安全性了。那么，接下来我们就介绍两种专门对恶意镜像进行扫描的工具：Anchore、dockerscan

* + 1. Anchore



与Clair不同，Anchore侧重于对镜像的审计，其有强大的对镜像的解析能力。

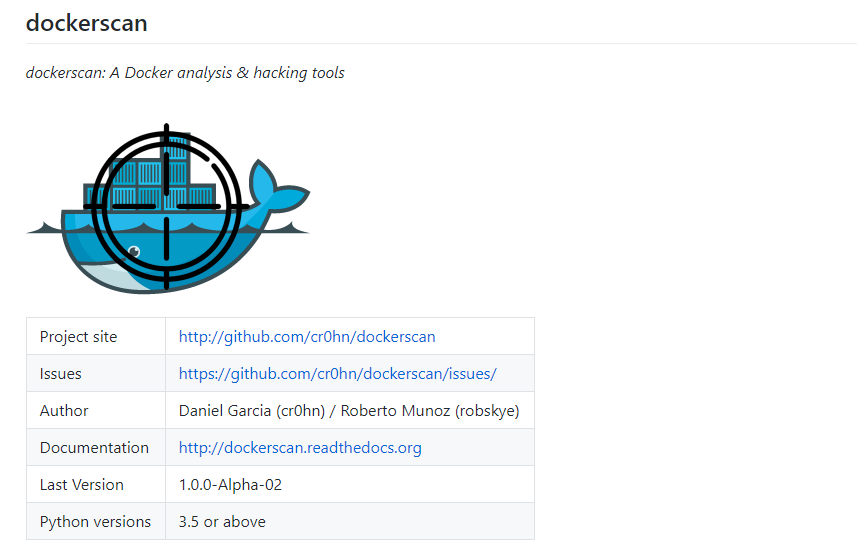




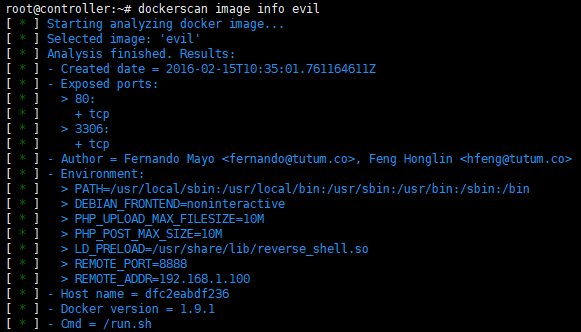
如图，Anchore可以通过审计输出恶意镜像和原始镜像之间的不同，比如：跟原始镜像对比，evil镜像上新增了wget和openssl等软件和一个后门文件 backdoor.apk.sh，那么我们就可通过这些信息推测出此镜像下载了wget然后从网上下载一个sh后门，执行恶意命令了。

Anchore就是通过审计这些特征来让用户判断是不是恶意镜像的

* + 1. Dockerscan



Dockerscan也是一款恶意镜像扫描的利器，其通过扫描镜像中的环境变量、操作命令以及端口开放信息来识别其是否为恶意镜像



我们通过几个环境变量(LD\_PRELOAD、REMOTE\_PORT、REMOTE\_ADDR)和运行run.sh的操作识别出evil镜像有一个连接C&C服务器的行为，从而断定这是一个恶意镜像

* 1. 镜像安全未来趋势

跟传统的杀毒软件类似，对镜像的安全检测将会从静态扫描发展到动态扫描，使用动态扫描工具(如：SELinux、Apparmor、Falco)监控容器里的操作，当容器里有异常的操作时，立即为用户给出警告，继而制止这种恶意行为，达到保证容器安全的效果。这样就省去了静态扫描遇到的时刻更新漏洞库和被绕过检测的麻烦。

1. 总结

镜像安全决定了容器安全，而目前Docker Hub上的镜像75%都存在漏洞，所以我们使用镜像运行容器前，一定要对镜像镜像进行扫描，从而提高安全性。

随着Gartner2017中EDR（Endpoint detection&response）技术的提出，基于机器学习的端点检测技术已经成为趋势。

每个容器中的行为单一（每个容器/OS一般只起一个应用），可以为检测减少很多噪声，所以将EDR技术运用于容器扫描扫描之中，将会得到不错的效果。镜像安全的解决方案也将会因EDR的兴起而变得多样化

参考资料：

Clair：<https://github.com/coreos/clair>

Anchore：<https://github.com/anchore/anchore>

Dockerscan：<https://github.com/cr0hn/dockerscan>