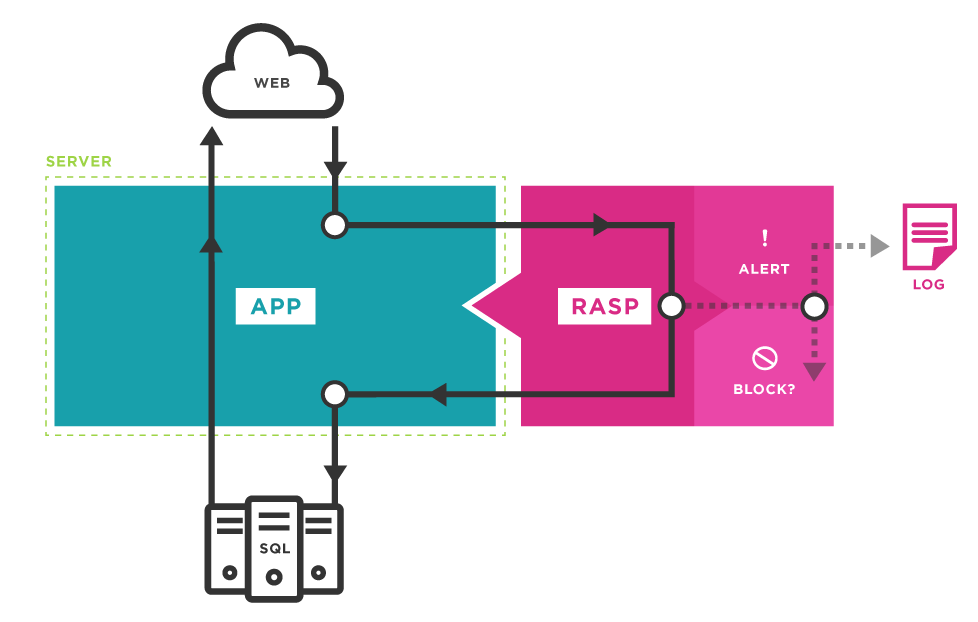
**0x01 什么是RASP**

RASP（Runtime application self-protection）运行时应用自我保护。Gartner 在2014年应用安全报告里将 RASP 列为应用安全领域的关键趋势，并将其定义为:

Applications should not be delegating most of their runtime protection to the external devices. Applications should be capable of self-protection (i.e., have protection features built into the application runtime environment).



rasp

RSAP将自身注入到应用程序中，与应用程序融为一体，实时监测、阻断攻击，使程序自身拥有自保护的能力。并且应用程序无需在编码时进行任何的修改，只需进行简单的配置即可。

**0x02 RASP能做什么**

RASP不但能够对应用进行基础安全防护，由于一些攻击造成的应用程序调用栈调用栈具有相似性，还能够对0day进行一定的防护。

除此之外，利用RASP也能够对应用打虚拟补丁，修复官方未修复的漏洞。或者对应用的运行状态进行监控，进行日志采集。

**0x03 WAF VS RASP**

传统的WAF主要通过分析流量中的特征过滤攻击请求，并拦截携带有攻击特征的请求。WAF虽然可以有效个过滤出绝大多数恶意请求，但是不知道应用运行时的上下文，必然会造成一定程度的误报。并且WAF严重依赖于特征库，各种花式绕过，导致特征编写很难以不变应万变。

RASP的不同就在于运行在应用之中，与应用融为一体，可以获取到应用运行时的上下文，根据运行时上下文或者敏感操作，对攻击进行精准的识别或拦截。于此同时，由于RASP运行在应用之中，只要检测点选取合理，获取到的payload已经是解码过的真实payload，可以减少由于WAF规则的不完善导致的漏报。

虽然RASP拥有WAF所不具有的一些优势，但是否能够代替WAF还有待商榷。毕竟WAF是成熟、快速、可以大规模部署的安全产品。两者相互补充，将WAF作为应用外围的防线，RASP作为应用自身的安全防护，确保对攻击的有效拦截。

**0x04 RASP的缺陷**

注入应用中的RASP虽然带来了获取应用运行时上下文的优势，但也带来了一定的缺陷：性能消耗。应用自身在进行常规运算的同时进行了一定的安全运算，造成了一定的性能消耗。不过根据Gartner 分析师的统计，RASP带来的性能消耗在5%~10%之间，在一定程度上仍然是可以接受的。

除此之外，由于RASP需要运行在应用中，不能像WAF一样在流量入口统一部署。需要根据应用开发的技术不同使用不同的RASP。比如.net应用与java应用需要不同的RASP产品，增加了部署成本。

下篇 <http://www.jianshu.com/p/53b50edb4a04>