ONE ASP RASP

Why use OneRASP ?

ONEASP Framework 为什么使用 OneRASP？

自内而外地保护应用程序

“现代安全技术无法为所有应用程序提供测试和保护。因此，应用程序必须具备自我检测、自我诊断和自我保护功能。这应当是 CISO（首席信息安全官）的首要任务。”

OneRASP——实时应用自我保护

近年来针对应用程序的攻击呈快速上升态势。对于大部分企业来说通过现有安全保护技术来保护应用程序不仅需要大量的人力、物力和财力，而且效果还不能得到保障，每个企业都迫切需要一种快速、简单有效的应用程序安全保护方案。Gartner 的分析师兼研究员 Joseph Feiman 敏锐的意识到这一点，于是他提出了“实时应用自我保护（Runtime Application Self-Protection）”的概念，也称为 RASP。Feiman 认为可以将漏洞扫描工具的程序测试功能和 WAF 的攻击拦截功能结合在一起，并将这些功能集成到应用程序中，这样应用程序就有能力在运行时对行为进行实时扫描并有能力进行实时拦截。就像人体疫苗一样让应用程序具备自我保护的免疫能力。

“根据设计思路，‘实时应用自我保护’（RASP）是向应用程序运行时环境添加保护功能，以保护应用程序”

现状：80% 的网络攻击直接针对应用程序

目前，在应用程序安全保护方面的投入还无法应对这一挑战。最近开展了一项研究，它以 400 多家大型组织在应用程序安全防护方面的做法为研究对象，找出了它们在应用开发部署过程中的“重大安全缺陷”：

△ 33% 从未检测应用程序是否存在安全漏洞

△ 50% 的应用程序安全防护预算不足

△ 65% 称，经常会因为客户的需求而使应用程序面临风险

△ 77% 将“急于发布”的压力作为程序代码中包含漏洞的首要原因

在开发过程中确保应用程序安全是非常艰巨的任务

△ 在软件开发周期中加入应用程序安全防护程序需要专业安全程序员和大量的时间

△ 你可能无法对每个企业应用程序都进行扫描 / 测试

△ 即使找到了漏洞，也可能缺乏足够的方法、能力或资源来修复

△ 瞬间完全修复漏洞再上线是不可能完成的任务

传统基础架构和边界保护技术无能为力

△ 基础架构和边界保护技术存在先天不足，它们不了解应用程序的逻辑与配置、事件与数据流、所执行的指令和数据处理，因此也就无法准确地检测出应用程序的漏洞，无法对抗应用程序级别的攻击

△ 防火墙对内部攻击无能为力，而这些攻击的破坏性不亚于外部攻击

△ 在云计算、移动互联时代，明显的边界已经越来越少。针对边界保护没有任何意义

AST 费力不讨好

很多公司使用 AST 动态扫描程序，希望在上线前将所有漏洞找出来并修复，这个愿望是美好的，从实践来看效果并不明显，因为这些工具通常比较复杂需要大量配置和很专业的人才，现代程序都是动辄几十万行，还会使用大量的第三方程序，所以分析并修复扫描出来的漏洞是非常费时费力的工作。同时漏洞更新速度慢，并不能确保所有漏洞被发现并修复。

WAF 不能完全满足需求

Web 应用程序防火墙是相对比较好的应用程序保护方案，但它的依据就是一些很简单的模式匹配，不会考虑输入内容是否将被传送给包含漏洞的代码。而一些攻击是需要了解应用程序的内部情况才能发现的，WAF 会漏过此类攻击。而且现在黑客翻墙技术非常高超，对于高水平攻击，WAF 形同虚设。RASP 框架不同于 WAF，它与要保护的应用程序结合在一起，根据上下文提供检测，在源代码级别为应用程序的易受攻击区域提供保护。RASP 就像是一种疫苗，即使恶意输入进入了内部，也能保护应用程序免受攻击。

为什么使用‘实时应用自我保护’（RASP）？

统计数字表明，你的应用程序中存在的漏洞，导致你面临着诸多风险，比如可能遭受跨站脚本攻击、SQL 注入攻击、恶意软件攻击及其他一些攻击。你的应用程序会被破坏，数据会泄露，这是很危险的。即使找到了漏洞，那也只是战斗的一半，漏洞的修补过程可能充满挑战，甚至是无法完成的任务。所以急需寻求一种有效的方式来保护你的传统应用程序、移动应用程序和 Web 应用程序。

△ 漏洞无处不在——要在开发过程中发现并修复每个漏洞，基本不现实，攻击者每天都在发现新的安全漏洞

△ 漏洞查找与修复——由于发布上线的压力，带病上线是常有的事情。漏洞还可能遗留在代码或者第三方代码里。通过修改代码来消除漏洞是需要时间的，在此期间，攻击者可以做非常多的事情

△ 直接保护它——不要把时间和精力浪费在传统安全保护方式：网络扫描过滤只是猜测，SIEM 也许会发现一些威胁但是无法阻止。请直接在应用程序里保护它

△ RASP 是很好的方式——OneRASP 在运行时扫描具体行为，在掌握上下文的基础上能很精确的识别并阻止攻击行为。无需修改代码，在数分钟内就能保护你的应用程序

△ 具备的优点——现有应用程序无需修改任何代码就可以在运行时进行自我保护，使用的漏洞规则集经过顶尖安全专家多年研究形成，能精确发现并阻止攻击行为。只针对关键保护区域进行扫描和保护，对 Java Web 系统性能影响极小

应用程序自我保护让应用安全保护更轻松

保护：

△ 拥有应用程序上下文，使攻击行为无处藏身

△ 对应用程序的逻辑和数据流进行运行时分析

△ 准确区分实际攻击行为和合法请求

洞悉：

△ 使攻击过程透明化，精确定位漏洞存在哪行代码或者那个 SQL 语句

△ 交互式仪表板，用于风险的优先级划分与补救

简单：

△ 安装过程轻松、快速，只需三个部署步骤，在几分钟内即可运行，提供保护

用 OneRASP 简化应用程序的安全防护

应用程序自我保护

△ 在数分钟内完成安装，为产品化应用程序提供保护

△ 确认并即时阻止攻击——准确、高效

△ 直接洞察，可以看到应用程序能够看到的一切

△ 消除臆测，准确定位代码行中的安全漏洞

补丁修复不管用

△ 很多程序只提供接口，无法得到代码，发现有漏洞自己也无法修复

△ 即使安全扫描器发现很多漏洞，修补这些漏洞也是非常耗时耗力的

△ 即使第三方软件商愿意修补漏洞，在不短的等待周期你无能为力，攻击行为可以为所欲为

△ 况且很多时候你根本就不知道存在哪些漏洞，使用运行时监控和保护程序非常必要

为什么选用 RASP？

△ 它工作在 Java 应用程序环境中

△ 它是一种被证明有效的运行时应用程序自我保护技术

△ 它是应用程序安全防护领域的佼佼者