云账号

云服务

云资产

# 云安全

## ****内容介绍****

本篇是系列文章的第二篇，分享一下自己对应用安全的理解。

## ****应用安全是什么****

应用安全的保护对象是应用系统，应用系统有各种表现形式，如Web、APP、小程序、客户端、系统等。应用安全就是围绕着这些应用的安全建设工作。

## ****应用安全需求来源与演变****

随着个人终端的普及以及互联网的发展，面向C端的/B端的 软件、应用、系统越来越多。

用户本身就对系统，应用就有安全性的要求。比如说window系统，office，来回中病毒肯定会骂微软，比如说游戏，游戏里全是外挂，直接不能玩，肯定就换游戏玩了。

然后就是通常状态下，特别是C端的应用，公司业务跟应用都呈现了非常大的绑定关系。

应用能够直接影响公司业务，所以应用的安全性，也越来越受到企业重视，其重要程度也往往排在安全方面的首要位置。

比如说所有的电商平台，要是有支付漏洞，或金额没校验等漏洞，能直接白嫖商品的，直接把公司干倒闭都有可能。

然后是随着互联网产业对人民的影响，国家层次上的好多基础服务业务也有了互联网应用。再然后因为黑灰产的危害，倒闭着国家立法去应对互联网犯罪行为，立法又倒闭着企业进行应用的安全建设。比如《网络安全法》规定了等级保护制度，国内所有系统都必须做等保。

## ****应用安全理念演变****

应用安全的理念，经历了从 SDL-DevSecOps 的演变过程。



微软是第一家提出了体系化的应用安全理念的公司。它于2004年提出了SDL模型。SDL围绕着软件生命周期，将安全性考虑融入到了整个软件开发过程，从多阶段多方面提高软件的安全性，降低了漏洞出现频率，拉低了软件修复成本。

因为其科学性与微软的背书，SDL一经发布就迅速成为了安全业内的标准指导思想，甚至有安全从业者将它用成了企业整体安全建设的指导思想。

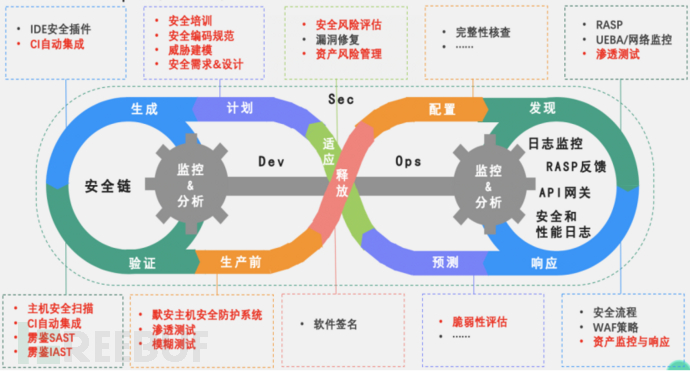
然而微软原始的SDL理念，是有适用范围的，是有适用条件的。它是微软基于其公司、产品、工作流、以及开发技术栈而产生的应用软件安全理念。当时微软的主要产品就是操作系统，其具有长周期性、一次性、变更少等特点，所以产生了SDL的理念内容。

SDL里，安全前置的工作非常多，而且非常重要。这是因为，操作系统这个产品，它必须前置那么多，如果不前置，如果出现问题，其安全修复成本就太高了。

SDL落地需要安全部门，从头就参与，一起走完完整的应用开发流程。操作系统这个产品的长周期性跟垄断性，决定了安全部门有足够的时间去做前置的工作。微软这个公司，也决定了他有完善的管理，完善的组织，完善的工作流程来让前置的安全工作落地。

所以SDL符合微软自身，但是不一定符合其他公司。这些区别特征注定了其他公司按照微软的标准实践去落地SDL是很难很难的。

所以后来出现了DevSecOps 这个应用安全理念。



因为市场上的主要产品、开发技术、运维技术等多方面的因素都随着时间发生了很多变化。

产品形态变化：没有公司染指操作系统了（除了乔布斯），产品从PC软件，到web系统到APP。开发时间可能只有操作系统的零头。应用越来越跟业务高度相关，本身就有快速更新，抢占市场的需求。

开发模型变化：瀑布模型，到敏捷模型，到devops模型。越来越快速，越来越敏捷，更新发布周期越来越短。

开发技术的变化：从C，到Java，到python，到GO。从MVC，到微服务，到云原生。其技术栈本身也添加了很多安全理念。技术也越来越解耦合，软件开发各阶段的漏洞修复成本拉平了。

运维技术栈变化：传统主机，到虚拟化，到云。从批量，到容器，到云原生，到CICD。

这些方面的变化与差别，共同导致了DevSecOps的出现。

这两种理念的内核都是一样的，都是围绕着应用的生命周期，结合现实的技术栈，现实的开发流程，从管理、技术、运营三个层面，来进行安全建设。SDL跟DevSecOps都只是这种内核的体现。

所以实际上是什么呢，看起来是理念从SDL变成了DevSecOps，实际上理念是不变的，就围绕着应用生命周期，在各阶段去做一些安全的事情。

实际中，要理解它的内核，不要本本主义，完全照搬，没有自己的思考，没有依据现实进行本地化。

就拿安全前置的工作来说，除了SDL里面的东西，我们还能做的有：

1. 应用的基础IT设施的架构如何设计
2. 应用涉及的数据，如何设计数据表，以便于数据安全建设
3. 应用的业务模式，功能设计，有没有业务安全风险，其风控应如何设置
4. 《网络安全法》规定了必须做等保，这个应该是几级，应该达到什么量级的时候做？

这里还要澄清一下从业人员可能出现的一种误解。SDL不能用作整个公司的安全建设理念，也不能用做整个应用安全方面的建设理念。它只能用作对单个应用进行安全建设的时候。

对整个应用安全方面进行安全建设，要有所有应用的统一管理，统一运营，SDL里并没有这方面的内容。

这种安全理念的误用现象，屡见不鲜，与之相似的还有零信任理念。还有就是不要本本主义，看见最佳实践就照搬，然后会因为公司条件不匹配，导致事倍功半，浪费大量人力物力，本来这些可以用来到别的安全方面，快速拉升整体安全水位的。

## ****应用安全的建设落地****

步骤：

应急跟需求：处理应急跟公司需求

应用梳理：梳理所有的应用，评估各应用的应用安全的高优先级内容。建设应用发现机制跟能力

状态调研及评估：调研安全建设状态，评估建设完成度。

决策工作内容：结合应用调研结果，结合公司整体的建设方案，以SDL内核为指导思想，来决定工作内容，内容顺序，落地方案。

整体应用安全方面的建设，可以从管理，技术，运营三个方面来做。

整体公司的应用管理要做好，最好有统一管理。消除应用黑洞，归属的黑洞。要建设应用的发现能力。

技术体系上，应用有什么类型，它们对应的技术栈，技术链路是不是都覆盖了，各种类型应用的安全重点分别是什么。

运营体系上，怎么囊括所有类型应用，他们的共性是什么，怎么自动化数据化运营。

具体到单个应用

按SDL理念有如下简化版的落地内容

立项阶段：理解这个产品，知道功能，面向C端还是B端。知道app，还是web，提出安全需求。目的：制作安全预案，方便后续工作展开。痛点：需要知道有项目在立项。

技术选型：安全开发框架选择，安全开发培训。比如Struts框架漏洞就很多。目的：提高安全水平，减少漏洞发生。痛点：开发语言太多，技术栈要求全面。

功能设计阶段：功能的安全设计及实现。目的：提高功能安全性。比如百度网盘的分享功能设计造成用户隐私泄露。痛点：不知道添加新功能。

开发阶段：代码审计。目的：白盒角度检查漏洞。0day都是代码审计发现的。痛点：开发语言太多，技术栈要求全面。

测试阶段：渗透测试。目的：黑盒角度检查漏洞。痛点：全面性，测试生产一致性。

发布阶段：发布行为控制与审计。目的：控制发版。比如拼多多测试卷自动上架生产环境，被大量薅羊毛。痛点：流程建设。

应用商城：加壳加固。没壳的APK，或客户端，破解起来太容易了。盗版问题。痛点：上架流程建设

持续运营：安全监控，日志审计。网马后门问题。痛点：系统建设。

大型的公司，可能同时有好多的项目在立项，还有好多已经上架的应用在持续更新。

它的工作量会非常大，所以运营工作对应的有 系统化，标准化，自动化的需求。

## ****应用安全与其他安全方面的耦合****

基础安全：

对应用关联域名、IP及服务器、数据库、OSS、配置文件等的关联关系。

运行环境安全，组件有问题导致应用不可用。例如F5漏洞。

办公安全：

后台支持类系统，相关工作流程的安全

行为控制及审计，日志记录

不安全的办公行为导致应用故障。各种误删库比如gitlab。内鬼泄密等。

数据安全：

密级数据与应用的关联关系

行为审计，日志记录，系统脱敏，数据加解密。

各种内鬼泄密，未授权泄密，网马泄密。

业务安全：

业务系统，相关业务流程的安全

风控管理，审计等。爬虫问题，功能设计问题。

## ****总结****

应用安全是围绕应用的

SDL是应用安全理念，不是整体公司安全理念

落地要灵活要适配，不要本本主义

## ****QA****

弟弟失业中，目前在上海，正在找工作。各位大佬如果对我感兴趣的话欢迎加我VX：qinbingjie1101