Cybellum

白皮书

安全的集成与发布

保障供应链及产品安全

Cybellum

简介

在过去的几年中，第三方组件及其代码已经从产品的次要因素转变为了产业当中的主要因素，其影响贯穿了整个软件开发周期及集成过程。

今天，大部分的企业或组织，通过供应链获得的软件产品，其实在网络安全性上是盲目的。 因为他们并不参与产品的实际开发过程，因而也无法访问该产品的所有源代码，就更无法知道软件产品的内部是什么？成分是什么？以及他们拥有的风险是什么？ 所以管理好供应链产品的内部构成，是该类企业能否获得成功的关键。

采用开发外包将会导致，应用程序的重要部分的行为在当今大多数流行的静态代码分析工具中无法被发现。 因为第三方软件，通常仅以二进制形式(被编译后)提供，在没有源代码的情况下，采用静态源代码分析工具进行检查，是无法完全分析解释出应用程序调用第三方代码的安全问题。

当软件模块被组装后

软件模块被组装的过程实际上和普遍设备被组装的过程是一样的.在软件生态系统中，构建者(或开发人员)创建软件产品，而购买者(或消费者)使用它们。 无论产品是移动应用程序，医疗设备，汽车ECU还是飞机固件，里面的软件程序也是通过采用不同的功能模块来组装成完整功能的商用软件。

代码的主要功能可能由公司自有的产品团队直接开发。 尽管如此在当今程序开发世界里大部分软件的功能都会或多或少的整合了第三方程序模块或组件，或在第三方程序模块或组件之上再基于特定的功能需求进行定制化开发。

第三方的组件通常不是由产品团队开发的，可能是使用了开源的组件，商业化的组件，甚至不同的部分也是由不同组织的团队开发的。 因而最终软件成品的代码将会由不同组织或开发团队的代码进行了组合。 确切的代码比例会因项目和行业而异，但几乎所有行业内的软件都以这种方式组装的。

第三方组件的集成形成了一个软件供应链，它同样具有传统制造业中供应链的所有挑战，以及许多网络安全风险。 但要基于手动管理该供应链的安全风险是基本不可行的，只有通过自动风险检测的解决方案才能最大限度地降低供应链中的风险。

Cybellum

组件存在风险

很多时候乍一看，该软件组件的功能实在太好用了，既可以获得所需的一些功能，又可以节省了自己编写代码的时间和金钱，但是其中存在的严重风险却经常被忽略，其实主要风险还是来自于漏洞，由于组件中的程序错误，使得攻击者有机会窃取重要信息，或导致产品的功能行为不正确或完全瘫痪，幸运的是已知的漏洞目前大部分都已经公布于众，多数都可以通过国际漏洞发布平台获得，如果您知道目前正在使用的某些组件是属于公共组件，那可以轻易的通过源代码找出已知漏洞，但是如果它是专有的模块组件，这样的做法是不可行的，没有商业组织愿意公布自己的源代码。

一旦您的产品使用了具有安全漏洞的软件组件，您和您的客户将面临严重的安全风险。 假设您销售的车辆，被黑客或非法攻击者利用其中某个第三方组件的漏洞来攻击您的汽车系统，导致严重的驾驶安全风险，危及驾驶员的生命，又或者您销售的医疗设备，由于某个第三方组件的漏洞可能会意外导致患者的死亡，即使最后没有危及生命，但其产生的影响也会让卖家的声誉遭受不可挽回的损害，进而造成无法估量的经济损失。

开发组织有时会侥幸认为那不是我的代码意味着不是我的问题，他们引入第三方软件组件的思路是假设它已经过其他人的质量和安全性测试，但这只是假设，可能测过也可能没测过，但无论如何，它仍然在最后的包装盒上留下了开发者的名字。 如果汽车因发动机密封部件故障而发生碰撞，供应链中的每个人都会遇到严重问责，从密封部件制造商到发动机制造商，以及汽车制造商本身，不论软件还是硬件，它们都具有与产品制造商和购买者所承载的完全相同的风险和依赖。 当开发组织发布包含第三方代码的产品时，尽管当中的每一行代码 ，包括所有第三方代码都是由他们负责开发的。 但由于以下种种原因，很多时候产品中的组件是不会做更新的：

- 开发人员没有意识到包含组件的安全性影响，他们关注的重点只是功能。
- 开发团队的漏洞及缺陷管理不完善无法跟踪修复，许多团队仅通过电子表格或简单列表来管理缺陷及漏洞，这样很难保持其准确性和更新速度。
- 没有人意识到组件存在，例如：为便携性而做的相关修改，或只是采用了静态链接的代码，是很容易被人遗漏的。
- 保持组件版本的最新状态很具有挑战性和难度，由于缺乏管理或资源。

Cybellum

攻击方式也在不断进化

安全研究人员或者黑帽子每天都会在软件组件中发现新的漏洞，即便你一直努力的研究和改进你的供应链，但你的产品也很难保证不包含已知漏洞的组件，目前已经有愈演愈烈的趋势。 例如，2015年，国家漏洞数据库(NVD)全年共有6,447个，而到了2017年，这一数字增加到14,712。

2018年，熔毁和幽灵漏洞在这个问题上引起了广泛的关注，这些都是我们设备中CPU(中央处理单元)的安全漏洞，它们允许恶意代码访问设备中在正常情况下是无法被访问的信息。 该漏洞公开披露后，厂家和采购商都纠结，是否应该快速有效地做出反应解决这个问题，因为其中涉及到大多数行业都缺乏供应链管理的问题，比如厂家也很想知道：

- 哪些版本的产品使用了易受攻击的程序组件
- 哪些系统上运行的产品包含了易受攻击的程序组件

其实解决这些问题并不难，只要通过适当的供应链安全流程和正确的技术，答案就在您眼前。 如果您已经采用了正确的供应链管理手段来把控风险，您完全可以可以快速，冷静的解决上述漏洞，也能够快速识别哪些资产正在使用受影响的组件。

供应链的安全管理

软件供应链安全管理是您业务的重要组成部分，您需要密切关注正在使用的组件，以确保它们适合您的产品，并且没有质量及安全风险。

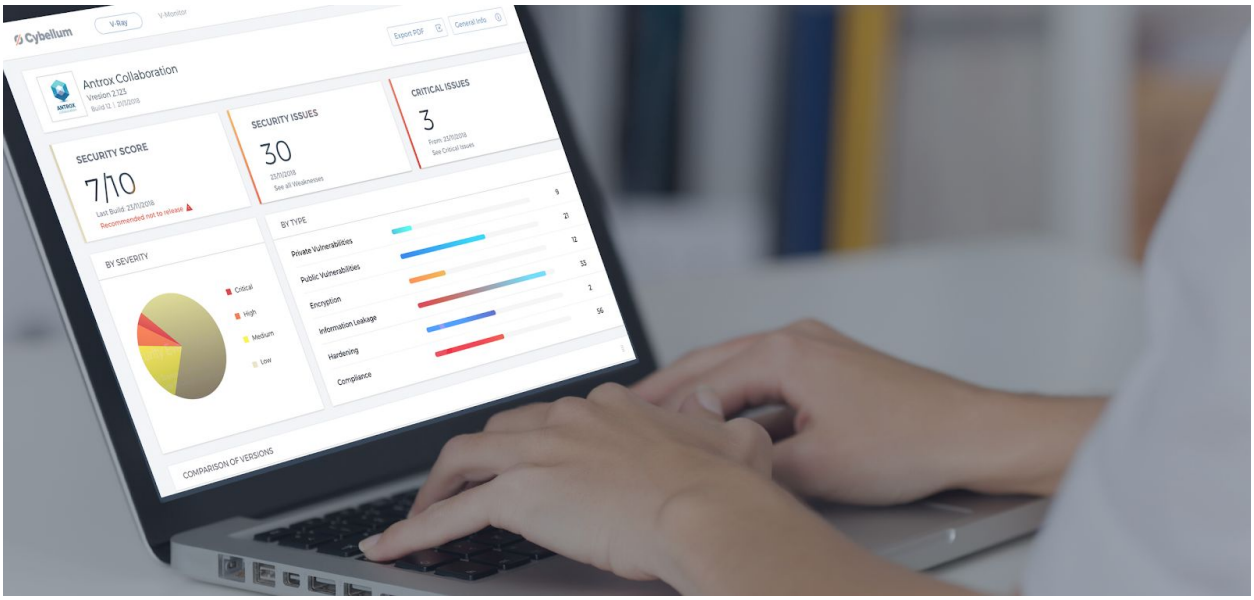
那您第一步需要做的是调整管控策略，例如：在任何情况下哪些组件都不容易受到攻击？有关第三方组件漏洞的政策是什么？什么时候向客户发布关键补丁？。 一旦定义好了策略，我们就能为引入第三方组件设置好管理门槛。

对于厂家而言，设置门槛基本在以下的位置：

1. 在发布期间，对产品中的组件进行全面的安全风险检查。
2. 在部署期间，监控供应链中已知和未知的漏洞，以便及时做出正确的风险决策，并更好的通过该信息来确定何时更新或替换这些组件，从而发布新的修正版本，并能及时的为客户提供恰当的更新帮助信息，以协助客户做出明智的更新决策。

工欲善其事，必先利其器

手动管理供应链的安全风险几乎是不可能的，但许多组织仍然使用电子表格或其他原始工具，手动跟踪安全风险，假如采用Cybellum的V-Ray平台，将关键任务自动化，真正使软件供应链管理成为可操作的实时流程。



首先，确定产品中的内容，使构建产品的所有组件变得透明可视化，有点类似于产品的成分列表。
一旦产品组件成分变得可视化，V-Ray平台就会自动执行分析，通过分析发现不同组件相对应的已知和未知的漏洞，让用户快速准确的了解产品的安全风险。
重要的是，V-Ray还会跟踪漏洞源，能够在第一时间发现针对不同组件而诞生的新漏洞或攻击技术，并向安全开发团队发出实时警报。

Cybellum

发布更安全的产品

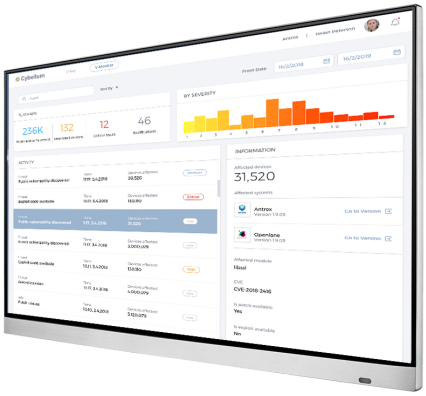
经过半个世纪的不可思议的创新和发展，软件行业不断成熟并发展成为一门真正的工程学科，虽然构建安全的软件以前是看起来是多么不可思议的工程，但现代软件工程流程已经完全考虑到了。 例如：DevSecOps流程，流程中的每个步骤和环节都确保了软件产品的安全性，健壮性和耐用性。

其中，供应链安全管理是构建安全软件产品的关键组成部分，采用有效的供应链安全管理，使得软件产品大大降低了安全风险，进而为每个人创造了一个更美好的世界。 软件开发厂商通过供应链管理，将更加清晰的了解其产品中的安全风险，而软件采购方也可以在采购周期中了解更加了解不同组件的安全问题，使得生产方不能在安全问题上有所松懈，必须最大限度地减少漏洞，确保产品的安全，而用户将最终受益于整个软件生态系统变得更安全，产品更可靠。

我们的产品

Cybellum V-Ray™

基于自动化的漏洞检测平台，提供完整的组件可见性和风险评估
利用自动反向工程扫描程序固件，并通过模仿黑客的攻击方法，高效的检测安全漏洞和威胁。
在无需提供源代码的自动化扫描后，V-Ray将清晰的展示所有集成组件的安全风险。 使得开发团队在及时评估组件风险后，准确的修复安全漏洞，并安全地发布产品。



Cybellum V-Monitor™

监控所有已部署的组件，让开发团队及时掌握公共，私有及暗源的新漏洞和威胁。

可集成到产品库存管理系统，通过自动化技术监控已部署的组件，当发现新漏洞时，平台将及时发出警报，让开发团队或运维团队在同一界面上就能及时掌握产品的安全风险。

以上两款产品都可以部署在云端和客户端服务器上，并可以作为SOC(安全运营中心)的一部分，完成企业对所有软件产品组件的漏洞检测，监控及风险图分析评估图。

产品技术

Cybellum的安全解决方案就像是一把漏洞检测的瑞士军刀，由前沿尖端的安全攻防技术组成，不断查找和验证全球最新的安全威胁。
我们检测漏洞的主要算法，是基于最新的机器学习技术，其工作方式如下：

学习阶段

在学习过程中，我们下载了数以千计的公开分类漏洞库，并合成了数百万个机密漏洞信息，从中提取了这些漏洞的所有可能性参数（比如堆栈调用的方法名称，不同的参数类型等等），让机器通过不同的样本学习，全面了解每类漏洞的发生的可能性，及相关测试方法。

通过长时间的学习过程之后，系统将得到了一个方法，当给定一个二进制文件时，可以快速生成我们在二进制文件中称为脆弱点的信息，这是二进制文件中具有高漏洞可能性的位置。

工作阶段

该技术由静态和动态两个主要部分组成：

- **静态** - 静态地浏览封装后的二进制文件，对其进行反汇编，以便找到公开可用漏洞的类别，程序脆弱点，构建问题等等。 通过不断的检测，算法学习，最终生成一组易受攻击的漏洞信息表。
- **动态** - 通过在类似“沙箱”一样的高度定制化执行环境中，执行二进制文件，目的是能够实时，准确的监控到被触发的程序漏洞，并实时通知告警，生成评估报告。

基于脆弱点的分析，同时包含压力测试等方法，系统每个新发现的漏洞或新的被公开漏洞，都会回到学习机器学习过程中用以改进算法。

Cybellum

检测漏洞类型包括：

公开的可被利用的漏洞

- 来自多个公共和专有数据库的漏洞
- 国家漏洞数据库(NVD)
- GitHub问题跟踪器
- 开源项目的错误跟踪器

内存损坏

- 缓冲区溢出-堆/栈
- 缓冲区读堆/堆栈
- 无效页错误
- 死锁
- 整数溢出
- 空指针引用
- 未初始化数据
- 同一块内存释放两次
- 内存释放后使用
- 内存越界
- 除零
- 类型混淆

安全配置错误

- 地址空间布局随机化(ASLR)
- 堆叠式粉碎保护器(SSP)
- 可执行空间保护(NX)
- 重定位表保护(RELRO)
- 栈金丝雀保护
- SAFESEH
- 数据执行保护(DEP)
- CFG
- 符号和剥离
- 强化源

危险的安全措施

- 最佳实践编码错误
- 错误处理问题
- 信息泄漏
    - 硬编码凭证
    - 纯文本密码
    - 哈希密码
    - 电子邮件,IP,URL,文件路径

- 加密
  - 可访问的加密密钥
  - 未加密的通信
  - 私人加密密钥
- 违反安全最佳实践
- 危险/不推荐使用的功能

许可证类型检测

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser"
- General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

支持的平台

- Linux
- Android
- QNX
- Windows
- NetBSD
- FreeBSD
- Proprietary
  RTOS

支持的架构

- x86
- x64
- ARM7
- ARM9
- ARM11
- MIPS
- PowerPC
- SuperH
- TriCore

Cybellum

关于Cybellum

Cybellum由IDF精英网络部门的专业人士创建，由攻击和防御安全专家组成，利用其丰富的军事经验来应对漏洞检测中最具挑战性的方方面面问题。