TLP:CLEAR

设计安全

改变网络安全风险平衡：

设计安全软件的
原则和方法

TLP:CLEAR

Communications Security Establishment

Canadian Centre for Cyber Security

Centre de la sécurité des télécommunications

Centre canadien pour la cybersécurité

目录

概述：易受攻击的设计 4
新增内容 6
如何使用本文档 7
设计安全 8
默认安全 9
对软件制造商的建议 9
软件产品安全原则 10
原则1：对客户的安全结果负责 11
解释 11
原则体现 14
原则2：奉行完全的透明度和问责制 20
解释 20
原则体现 21
原则3：由高层领头 26
解释 26
原则体现 27
设计安全的策略 28
默认安全的策略 30
加固指南与宽松指南 32
对客户的建议 33
免责声明 34
资源 35
参考文献 36

概述：
易受攻击的设计

随着面向互联网的系统越来越多地将我们连接到直接影响我们经济繁荣、生计甚至健康的关键系统，技术已经融入到人们日常生活的几乎所有方面，其范围包括个人身份管理到医疗保健等。这种便利有其缺点，举一个例子，网络漏洞已在全球范围内导致医院取消手术并转移患者护理。关键系统中的不安全技术和漏洞可能会招致恶意网络入侵，从而导致潜在的安全[1]风险。

因此，对于软件制造商来说，将设计安全和默认安全作为产品设计和开发流程的重点是至关重要的。一些供应商在软件保障方面取得了巨大的进展，推动行业向前发展，而有些供应商则持续落后。编写组织强烈鼓励每个技术制造商在构建其产品时基于减轻客户的网络安全负担的理念，包括无需不断对其系统进行监控、例行更新和损坏控制来减轻网络入侵的影响。我们也敦促软件制造商以一种促进配置、监控和例行更新自动化的方式构建其产品。我们鼓励制造商承担起改善其客户安全结果的责任。过去，软件制造商一直依赖于修复客户部署产品后所发现的漏洞，并要求客户自费应用这些补丁。只有采纳设计安全实践，我们才能打破不断创建并应用修复补丁的恶性循环。**注意：**“设计安全”这一术语包含设计安全和默认安全两方面。

为了实现这种高标准的软件安全，编写组织鼓励制造商优先考虑将产品安全性的融入作为功能和上市速度的关键先决条件。随着时间的推移，工程设计团队将能够建立一种新的稳态节奏，其中安全性真正融入设计之中，并且维护工作量更少。

与这一观点不谋而合的是，欧盟在《网络弹性法案》中强调了产品安全的重要性，强调制造商应在产品的整个生命周期实施安全措施，以防止制造商将易受攻击的产品引入市场。

[1] 编写组织认识到“安全”一词根据其上下文具有多种含义。在本指南中，“安全”是指提高技术安全标准以保护客户免受恶意网络活动的侵害。

为了创造一个技术和相关产品对客户来说更安全的未来，编写组织敦促制造商改进他们的设计和开发计划，只允许交付设计安全和默认安全的产品。在开发之前，设计安全的产品在概念阶段就以客户安全为核心业务目标，而不仅仅是技术功能。设计安全的产品在开始开发之前就以该目标为出发点。现有产品可以经过多次版本更新逐渐演变到设计安全的状态。默认安全的产品是那些“开箱即可安全使用”的产品，几乎或完全不需要更改配置，并且无需额外费用即可拥有安全功能。若将这两个理念相结合，那么保持安全的大部分责任就被转移给了制造商，并降低了客户成为因配置错误、客户打补丁速度不够快或许多其他常见问题导致的安全事件受害者的可能性。

美国网络安全和基础设施安全局（CISA）、美国国家安全局（NSA）、美国联邦调查局（FBI）和以下国际合作伙伴[2]在本指南中提供了建议，作为软件制造商确保其产品安全的路线图：

- 澳大利亚网络安全中心（ACSC）
- 加拿大网络安全中心（CCCS）
- 英国国家网络安全中心（NCSC-UK）
- 德国联邦信息安全办公室（BSI）
- 荷兰国家网络安全中心（NCSC-NL）
- 挪威国家网络安全中心（NCSC-NO）
- 新西兰计算机应急响应小组（CERT NZ）和新西兰国家网络安全中心（NCSC-NZ）
- 韩国互联网振兴院（KISA）
- 以色列国家网络安全指导委员会（INCD）
- 日本国家网络安全事件准备和战略中心（NISC）以及日本计算机应急响应小组协调中心（JPCERT/CC）
- OAS/CICTE政府网络安全事件响应团队（CSIRT）美洲网络
- 新加坡网络安全局（CSA）
- 捷克共和国国家网络与信息安全局（NÚKIB）

编写组织承认许多私营部门的合作伙伴在推进设计安全和默认安全方面所做的贡献。本文档旨在推动有关关键优先事项、投资和决策的国际对话，以实现技术在设计上和默认情况下安全、可靠和有弹性的未来。为此，编写组织向有关各方征求对本文档的反馈，并打算召开一系列反馈会，以进一步完善、明确和推进我们的指南，从而实现我们的共同目标。

有关产品安全重要性的更多信息，请参阅CISA的文章，《不安全技术的代价以及我们可以采取的措施》。

[2] 以下称为“编写组织”。

新增内容

本报告的首次发布在软件行业内引发了大量讨论。每天都有关于组织和个人遭受攻击的新闻，突显了对如何解决软件产品中的长期和系统性问题进行更多讨论的必要性。

自2023年4月发布以来，编写组织（以下简称“我们”和“我们的”）收到了来自数百个个人、公司和行业协会的深思熟虑的反馈。反馈中最常见的请求是对三个原则提供更多信息，因为这些原则既适用于软件制造商，也适用于他们的客户。在本文档中，我们对原始报告进行了扩展，并涉及了其他主题，如制造商和客户规模、客户成熟度以及这些原则的适用范围。

软件无处不在，没有任何一份报告能够充分涵盖软件系统、软件产品开发、客户部署与维护，以及与其他系统集成的所有内容。对于下方未明确对应特定环境的指南，我们期待听取社区的意见，了解本文件描述的实践如何带来特定的安全改进。

这份报告也适用于人工智能（AI）软件系统和模型的制造商。虽然它们可能与传统形式的软件有所不同，但基本的安全实践仍然适用于AI系统和模型。一些设计安全实践可能需要修改，以考虑专门针对AI的因素，但三个总体的设计安全原则适用于所有AI系统。

我们认识到，对软件开发生命周期（SDLC）做出改变，使其符合这些设计安全原则并非一项简单的任务，可能需要一些时间。此外，许多这些建议对于小型软件制造商来说可能难以实施。我们认为，软件行业需要广泛提供使产品更安全的工具和程序。随着更多人和组织将注意力集中在软件安全的改进上，我们相信有创新的空间来缩小大型和小型软件制造商之间的差距，从而使所有客户受益。

本次对初版设计安全报告进行的更新是我们与众多相互关联的利益相关者社区建立合作伙伴关系的承诺的一部分，这些利益相关者社区支撑着我们的技术生态系统。本次更新是我们从该生态系统的众多部分听取反馈的结果，我们将继续倾听不同的观点并从中学习。尽管前方存在许多挑战，但随着我们对已经采用设计安全理念并常常取得成功的个人和组织有更多的了解，我们对未来感到非常乐观。

如何使用本文档

我们敦促软件制造商遵守本文档中的原则。软件制造商可以按照下列步骤，通过公开记录他们所采取的行动来表明他们的承诺。我们鼓励软件制造商找到符合这一原则之精神的策略，并创建工件（artifact）来提供令人信服的案例，从而说服持怀疑态度的当前和潜在客户，以此证明他们体现了设计安全的理念。

除了软件制造商应采取行动之外，客户也可以利用本文档。购买软件的公司应该从遵守本文档中所列原则的例子中汲取灵感，向他们的供应商提出尖锐的问题。通过这样做，客户可以帮助推动市场朝着设计更安全的产品方向发展。客户可以向供应商提出的问题的一个例子见CISA的《K-12技术购置指南》。

我们鼓励企业客户将这些实践纳入采购流程、供应商尽职调查评估、企业风险接受决策以及在评估供应商时采取的其他步骤中。客户还应督促他们的供应商公开记录每个供应商采取的设计安全措施。所有这些共同作用，可以产生对安全有强烈需求的信号，从而鼓励并帮助软件制造商采取措施提高安全性。换句话说，正如我们寻求在软件制造商中创造普遍的设计安全理念一样，我们需要在他们的客户中创造一种“需要安全”的文化。

设计安全

“设计安全”意味着技术产品的构建方式可以合理地防止恶意网络参与者成功访问设备、数据和连接的基础设施。软件制造商应开展风险评估，以识别和列举关键系统所面临的普遍网络威胁，然后在产品蓝图中包含保护措施，以应对不断变化的网络威胁形势。

我们还建议采用安全信息技术（IT）开发实践和多层防御（称为纵深防御），以防止恶意参与者危害系统或未经授权访问敏感数据。编写组织进一步建议制造商在产品开发阶段使用定制的威胁模型来解决对系统的所有潜在威胁，并考虑每个系统的部署过程。

编写组织敦促制造商对其产品和平台采取全面的安全方法。设计安全的开发需要软件制造商在产品设计和开发过程的每一层都战略性地投入专门的资源，而这些资源不能在以后“附加”。它要求制造商高层业务主管发挥强有力的领导作用，将安全作为业务优先事项，而不仅仅是一项技术功能。业务领导人员和技术团队之间的这种协作要从设计和开发的初步阶段一直延伸到客户部署和维护阶段。我们鼓励制造商做出艰难的权衡和投资，包括那些对客户“不可见”的权衡和投资（例如，迁移到消除普遍存在的漏洞的编程语言）。他们应该优先考虑保护客户的功能、机制并实施保护客户的工具，而不是那些看起来很有吸引力但会扩大攻击面的产品功能。

没有任何单一的解决方案可以结束由恶意网络参与者利用技术漏洞所带来的持续威胁，而且“设计安全”的产品也将继续受到漏洞的影响；但是，大量的漏洞是由一部分相对较小的根本原因造成的。制造商应制定书面路线图，以使其现有产品组合与更多设计安全实践保持一致，确保仅在特殊情况下才偏离这种实践。

编写组织承认，对客户安全结果负责并确保这种级别的客户安全可能会增加开发成本。但是，在开发创新技术产品和维护现有产品的同时，向设计安全实践投资可以显著改善客户的安全态势，并降低遭受攻击的可能性。设计安全原则不仅可以增强客户的安全态势和开发人员的品牌声誉，而且这种实践从长远来看还可以降低制造商的维护和修补成本。

下方列出的“对软件制造商的建议”部分提供了供制造商考虑的产品开发实践和政策清单。

默认安全

“默认安全”意味着产品开箱即可抵御普遍存在的攻击技术，无需额外付费。这些产品可以抵御最普遍的威胁和漏洞，而最终用户无需采取额外的措施来保障其安全。默认安全的产品旨在让客户敏锐地意识到，当他们偏离安全的默认设置时，除非他们实施额外的补偿控制措施，否则他们会增加受攻击的可能性。默认安全是设计安全的一种形式。

- 安全配置应该是默认基线。默认安全的产品会自动启用所需的最重要的安全控制措施，以保护企业免受恶意网络参与者的攻击，并且可以在无需额外费用的情况下使用和进一步配置安全控制措施。
- 安全配置的复杂性不应是客户的问题。组织的IT员工经常负担过重的安全和运营责任，从而导致只有有限的时间来理解和处理安全隐患和实施缓解措施，而这些是实现稳健的网络安全态势所需的工作。制造商可以通过优化安全产品配置 —— 确保“默认路径”的安全 —— 确保他们的产品按照“默认安全”标准安全地制造、分销和使用，从而帮助他们的客户。

“默认安全”产品的制造商不会为实施额外的安全配置收取额外费用。相反，他们将这些安全配置包含在基础产品中，就像所有新车都包含安全带一样。

安全不应是奢侈的选择，而应被视为客户无需谈判或支付更多费用即可享有的权利。

对软件制造商的建议

这份联合指南为制造商制定书面路线图以实施和确保IT安全提供了建议。编写组织建议软件制造商实施以下部分中概述的策略，以通过设计安全和默认安全的原则来对客户的安全结果负责。

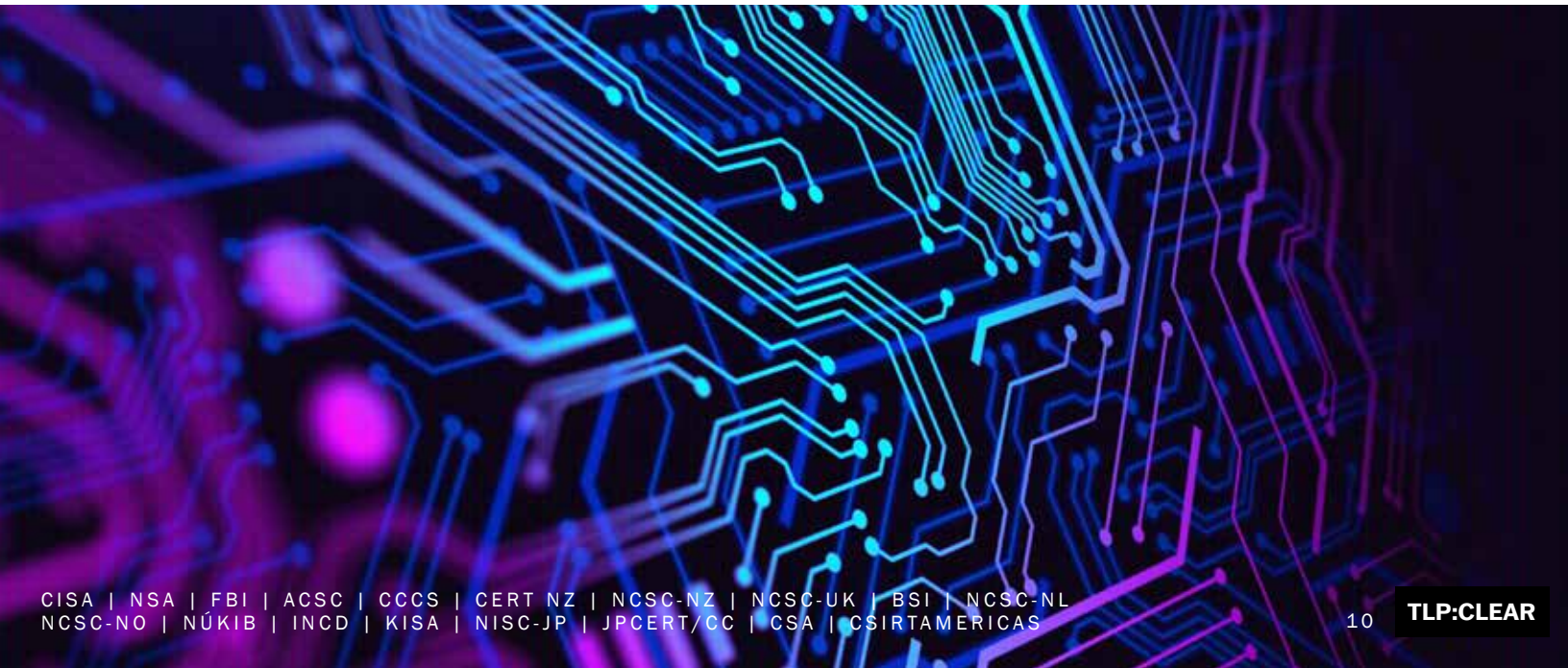
软件产品安全原则

我们鼓励软件制造商采取优先考虑软件安全的战略重点。编写组织制定了以下三个核心原则，以指导软件制造商在开发、配置和交付其产品之前将软件安全构建到其设计过程中。

1 **对客户的安全结果负责**并相应地开发产品。安全的责任不应该只落在客户身上。

2 **奉行完全的透明度和问责制。**
软件制造商应该提供安全可靠的产品，凭借自己的这一能力在制造商社区中与众不同，并对此感到自豪。这可包括分享他们从客户部署中学到的信息，例如默认采用强大的身份验证机制。这也包括坚定承诺确保漏洞警报和相关的常见漏洞和曝光（CVE）记录完整、准确。然而，要警惕将CVE视为负面指标的诱惑，因为这些数字也是健康的代码分析和测试社区的标志。

3 **建立组织结构和领导力来实现这些目标。**
虽然技术主题方面的专业知识对于产品安全至关重要，但高层管理人员是在组织中实施变革的主要决策者。高层管理人员需要在组织范围内将安全优先作为产品开发的关键要素，并与客户建立合作伙伴关系。



为了实现这三个原则，制造商应该考虑几种运营策略来改进他们的开发流程。

与公司高层领导召开例行会议，以便在组织内推动对设计安全和默认安全的重视。应制定政策和程序来奖励开发遵守这些原则的产品的生产团队，这可以包括对实施出色的软件安全实践提供奖励，或晋升阶梯和晋升标准方面的激励措施。

围绕软件安全对业务成功的重要性开展工作。例如，考虑指定一名“软件安全负责人”或一支“软件安全团队”，由他们专门负责维护直接将软件安全标准与制造商责任联系起来的业务和 IT 实践。制造商应确保他们的产品拥有稳健、独立的产品安全评估和评价程序。

在资源分配和开发过程中使用定制的威胁模型来优先考虑最关键和影响最大的功能。威胁模型需考虑产品的特定用例，并使开发团队能够强化产品。最后，高层领导人员应该让团队负责交付安全的产品，将其作为产品卓越性和质量的关键要素。

作为本指南2023年10月更新的一部分，我们通过以下解释、体现和证据对这三个原则进行了扩展。

原则1：对客户的安全结果负责

解释

现代最佳实践要求软件制造商投资于产品安全工作，包括**应用程序加固**、**应用程序功能**和应用程序**默认设置**。

软件制造商需要通过使用流程和技术来提高有意破坏应用程序的恶意参与者的成本，以此实施**应用程序加固**。应用程序加固协议和程序有助于使产品抵御来自智能恶意参与者的攻击。像加固、产品安全和韧性这样的术语都与产品质量密切相关。我们推崇的理念是安全性必须是“内置的”，而不是“附加的”。[1]通过在软件中内置安全性，软件制造商不仅可以提高客户的安全性，还可以提高其产品质量。示例的策略包括确保用户输入经过验证和清理，不直接输入到代码中（例如，使用参数化查询），使用内存安全编程语言，严格的软件开发生命周期（SDLC）管理，以及使用硬件支持的加密密钥管理。

应用程序需要支持与网络安全相关的**应用程序功能**。这些功能有时被称为“能力”，它们以有助于维护或提高客户安全态势的方式扩展产品或服务的功能性。

与安全相关的功能示例包括支持所有网络连接的传输层安全（TLS）、单点登录（SSO）支持、多重身份验证（MFA）支持、安全事件审计日志记录、基于角色的访问控制（RBAC）以及基于属性的访问控制（ABAC）。

这些产品功能中的一些功能是可配置的，使客户能够更轻松地将产品集成到其现有环境和工作流程中。这些配置意味着在客户进行配置之前，应用程序必须设好**默认设置**。这些默认设置需要设成“开箱即可安全使用”，以便客户投入更少的资源来使其众多技术产品更加安全。

其中的每个要素——应用程序加固、应用程序安全功能和应用程序默认设置——都对应用程序的安全性以及所产生的客户安全态势发挥着作用。软件制造商应该考虑其中的每个要素以及它们之间的关系。制造商应该考虑的不仅仅进行投资来将这些要素纳入其产品中。制造商应该更进一步考虑这些要素如何改变其客户的实际安全态势，是使其变得更好还是变得更坏。

制造商应该对其客户的安全结果负责，而不仅仅是衡量他们自己的努力和投资。责任应该放在上游，即制造商那里，这样才最有可能降低受攻击的几率。

令人遗憾的是，当今的情况并非如此。太多的制造商将安全的责任转嫁给客户，而不是投资于全面的**应用程序加固**。例如，当制造商修补一个漏洞时，我们经常看到类似的漏洞暴露出来，因为他们解决的是症状，而不是造成该缺陷的根本原因。产品可能在代码库的不同部分针对相同类型的漏洞实施不同的缓解措施。举一个典型的例子，在制造商修复了一个输入清理漏洞后，研究人员或攻击者发现代码路径无法受益于改进的输入清理。制造商对漏洞逐个进行修复，而不是统一代码库以消除整个应用程序中的该类漏洞。

应用程序功能既可以为客户带来好处，也可能带来风险。允许与许多外部系统和版本整合的功能可以大大增加产品的价值。然而，如果支持没有退出计划的功能（例如网络协议），在客户不了解持续使用该功能会有什么影响的情况下，可能会让客户容易受到攻击。例如，一些产品仍然使用源于1990年代或2000年代的网络协议，而我们现在已经知道这些协议不安全。有许多因素可能减缓客户升级和部署现代安全措施的速度。他们可能使用与组织网络的其余部分整合的产品，但缺乏现代安全措施，阻碍了IT团队的现代化进程。尽管如此，软件制造商仍然可以将这些模式纳入他们的规划过程中，以鼓励客户保持最新状态。

应用程序默认设置是给客户带来潜在风险的又一个领域。制造商通常会选择某些默认设置，使客户更容易使用他们想要的应用程序功能。这种做法的缺点是，对于可能不需要某些默认启用的功能和协议的客户来说，它增加了攻击面。此外，许多安全控制措施默认处于关闭状态，或者需要客户花时间配置其设置来提高安全性。明确的威胁建模是可能有助于确定哪些功能应默认开启，或者哪些设置需要默认安全的一种策略。另一种策略是研究使功能更容易被管理员发现的方法。

一些制造商交付的产品会带有可能对部分或所有客户造成风险的默认值。他们通常选择制作一份**加固指南**，客户必须自费实施该指南，而非设置更安全的默认值。加固指南存在几个常见的问题。有些加固指南很难找到，并且没有得到良好的支持。还有些加固指南实施起来较为复杂，有时需要进行软件开发来编写扩展模块。尽管如此，有些人仍然假设读者拥有丰富的网络安全经验，了解各种设置以何种方式改变攻击面。对于那些对攻击者的工作方式不完全了解的从业人员来说，他们可能无法正确实施加固指南的指示，尤其是在这些指示没有清晰说明利弊权衡的情况下。此外，并非所有加固指南都是由对攻击者的策略和经济效益有深刻了解的工程师编写的，这导致他们创建的加固指南即使被忠实实施，也是无效的。数百万客户正承担着对多个软件或系统实例进行加固的责任，而且这通常是在资源有限的环境中完成的。依赖加固指南根本不足以保障安全。

一个应用程序的设置应该针对制造商对威胁环境的当前理解持续评估，无论这些设置是默认的还是由客户设置的。应用程序应该有明确的指标，以表明这些设置可造成的潜在风险，并应该让这些指标为人所知。就像现代汽车有安全带指标一样，如果您试图在未系好安全带的情况下驾驶，就会发出警报来表明该指标，软件也应该表明关于系统安全状态的指标。如果一个应用程序配置为不要求管理员账户使用多重身份验证（MFA），则应定期提醒管理员，如果他们不配置MFA，那么他们及其整个组织都会面临危险。此外，如果一个应用程序配置为支持目前已知实施弱加密的旧协议，则应定期向管理员明确表明组织处于危险之中，并提供解决这种情况所需的资源。我们敦促制造商实施产品内置的定期提示，而不是依赖管理员的时间、专业知识和意识来理解加固指南。在安全性和可用性考量之间取得平衡，在此方面显然存在着创新的机会。

上述每个要素都会造成一种难以维持的局面，即客户需要研究、投资、购买、配备人员、部署和监控额外的**安全产品**来降低受到攻击的可能性。中小型组织（SMO）通常无法做出这些选择。他们面临专业知识、资金和时间的匮乏，这会对带宽和功能造成压力，迫使安全性的优先级别降低，在总体上加剧集体风险。相反，相对较少的制造商会扩大对安全性的投资。一言以蔽之，这个问题就是，软件行业需要更安全的产品，而不是更多的安全产品。软件制造商应引领这一转型。

> ***软件行业需要更安全的产品，而不是更多的安全产品。软件制造商应引领这一转型。***

如今，我们有时会读到制造商的评论，解释说由于客户未启用特定的安全功能或未遵循具体的加固指南，从而受到了攻击。制造商应该反其道而行之，在发生攻击后应解释某一特定的安全功能或具体的加固指南是否本可以防止该攻击，并考虑免费将其设为默认设置。在那些产品本身在设计和实施阶段未经充分加固的情况下，制造商应解释他们如何努力消除其产品系列中的此类漏洞。

软件制造商有责任确保其产品在设计和开发过程中将安全视为最高优先事项。为此，他们应该**客观地衡量他们在该领域工作的结果**。我们呼吁制造商不仅要专注于其内部工作，还要客观地衡量并定期报告产品安全工作和配置的结果和成效，并建立一个反馈回路，对软件开发生命周期（SDLC）带来改变，从而实现客户安全方面的可衡量改进，并创造更安全的产品。报告应包括匿名数据，学术界和研究安全性的社区可以使用这些匿名数据来跟踪高水平趋势并衡量整个生态系统范围内的进展。

原则体现

软件制造商和在线服务机构应设法表明他们在实施这一原则方面取得的成功。他们应寻求提供以工件（artifact）形式的证据，供外部人员检查。没有任何单一的工件能够证明制造商正在实施稳健的设计安全计划，但通过提供各种工件，他们就可以说明制造商致力于开发安全产品。这种方法符合“以实际行动示人，而非空谈”的精神。

为了体现这一原则，软件制造商应考虑采取下列措施。编写组织认识到，很少有软件制造商能够在设计安全的旅程开始时就立即实施这些实践并产生相应的工件。此外，软件制造商将需要根据客户在实地部署产品的方式来考虑此列表的优先等级，以实现最大的安全效益。

默认安全原则

1. **消除默认密码。**默认密码持续成为每年许多攻击的起因。承诺消除这一长期存在的问题将阻止攻击者轻易获得访问权限。同样，制造商应考虑应该实施哪些密码管理实践，例如最小密码长度和禁用已知泄露的密码。

2. **进行实地测试。**随着技术不断发展并变得更加复杂，对于软件制造商来说，进行以安全为中心的用户测试变得越来越重要，以便了解其产品在实地的安全态势。类似于用户研究为软件开发需求提供信息，软件制造商还应进行以安全为重点的用户研究，以了解安全性方面的用户体验（UX）存在哪些不足之处。通过观察客户在现实环境中如何部署和使用其产品，软件制造商可以获得对其安全功能和控制措施的可用性和有效性的宝贵见解。这些见解可以帮助识别需要改进的领域，并优化其产品以更好地满足客户对安全性的需求。例如，实地测试可能会建议在用户体验（UX）流程、默认设置、警报和监控方面进行更改。实地测试还可以显示过去在产品设计上的改进如何降低安全补丁的速度，减少配置错误，并最大程度减小攻击面。

制造商应考虑以下问题：

- 客户是否正确实施了加固指南？
- 产品现有的安全功能在实地是否表现如预期？
- 这些功能是否能实际抵御现实世界的攻击？
- 哪些功能可以更好地降低遭受攻击的可能性？

注意：为了更深入了解这些要素，软件制造商可能希望与客户合作进行红队演练，以查看产品如何抵御攻击。这些实地测试可以在客户的实际现场进行，在网上进行，或者 以保护隐私的方式通过应用程序进行遥测。

3. **缩小加固指南的大小。**制造商可以通过简化甚至消除产品加固指南，并专注于客户在部署产品时应优先考虑的最关键安全措施，从而提升客户的安全态势。制造商应该确定其产品最容易遭受的安全风险，并就如何减轻这些风险提供清晰简明的指导，而不是让客户面对一长串的安全措施清单而无所适从。此外，制造商应为客户提供工具和自动化操作，以简化实施安全控制措施的过程，例如可以轻松在其环境中部署的脚本。这些工具还应能够验证并清晰地显示对原始基准所做的更改。通过简化加固指南并为客户提供易于使用的工具和自动化操作，制造商可以减轻客户的负担，并帮助确保其产品以安全的方式部署。一种策略是考虑实施帕累托法则（Pareto principle），减少常见用例（80%）的步骤数，然后为较不常见的场景（20%）提供情境指导和工具。通过这种方式，软件制造商能够将简单的事情变得简单，将困难的事情变得可能。实地测试将成为一个强大的工具，能够衡量客户发现、理解和实施加固指南需要多长时间。制造商应考虑如何在产品内部提示管理员采取行动，而不是依赖于他们来实施加固指南中的任务。

4. **积极阻止使用不安全的遗留功能。**

通过明确的升级路径优先考虑安全性，而不是向后兼容性。发布博客文章，展示采用更安全的功能和协议，并通过公告（可能在产品内部发布公告）弃用不安全的功能。相当多的客户已证明，他们不会让自己的系统与现代网络、身份和其他关键安全功能保持同步。在某些情况下，客户担心升级会导致现有功能出现故障。通过尽可能使升级无缝完成，客户可能会更频繁、更快速地升级，并获得安全修复。软件制造商应积极提示客户沿着升级路径进行升级，从而降低客户风险。

5. **实施引人注意的警报。**

类似于汽车中未系安全带时不断发出声音的安全带提示音，制造商应在用户或管理员处于真正不安全的状态时及时、反复发出警报，警告管理员他们在其环境中使用了已弃用的协议，并建议升级路径。在用户、管理员或应用程序配置处于不安全的状态时，及时、反复发出警报。定期明确将不安全模式告知管理员。可以增设一个功能，要求超级管理员在每次登录时确认其账户未设置多重身份验证（MFA），甚至禁用某些关键功能，直到启用MFA为止。还有创新的空间来实现这些目标，同时避免因过多的警报而产生疲劳。

6. **创建安全配置模板。**

这些模板可以根据组织的风险偏好，将某些配置预设成安全设置。尽管使用低/中/高安全模板可能过于简单，但这个例子说明了可以对许多设置进行更新来管理组织的风险。制造商对已识别风险的加固指南可以为模板提供支持。

安全产品开发实践

1. **记录符合安全的软件开发生命周期（SDLC）框架的情况。**安全的SDLC框架提供了涵盖人员、流程和技术的目标和示例。考虑对已实施的安全SDLC框架控制措施发布详细的描述，并描述已使用的任何替代控制措施。在美国境内，考虑使用NIST的安全软件开发框架（SSDF）。虽然SSDF不是一份检查清单，但它“描述了一套用于安全软件开发的基本、可靠的实践”。

2. **记录符合网络安全绩效目标（CPG）或同等目标的情况。**当一个组织声明他们符合NIST的SSDF标准时，他们即断言他们的SDLC以众所周知的最佳实践为指导。然而，仅拥有稳健的SDLC是不够的。他们还需要保护自己的企业和开发环境免受恶意参与者的入侵，这些恶意参与者会在产品仍在开发时就试图操纵产品的安全属性。这不是一种理论上的攻击，而是已经发生的攻击，并对客户造成了不利影响，进而影响到国家安全。组织应考虑发布关于其符合CISA的CPG、NIST网络安全框架（CSF）或其他网络安全计划框架的详细信息。

3. **漏洞管理。**一些制造商有一个漏洞管理计划，专注于修补内部或外部发现的漏洞，但仅此而已。更成熟的计划包括对漏洞及其根本原因进行广泛的基于数据的分析，采取措施来系统性地消除整个类别的漏洞[3]。他们围绕制定质量规划、质量控制、质量改进和质量测量实施正式的计划。他们将缺陷管理视为一个业务问题，而不仅仅是一个安全问题。这些计划在某些方面与其他行业的质量和安全计划没有什么不同。

4. **负责任地使用开源软件。**使用开源软件时，应负责任地审查开源软件包、促进向依赖项贡献代码，并帮助维持关键组件的开发和维护。作为参考，日本经济产业省（METI）已发布《关于利用OSS并确保其安全性的管理方法的用例示例集》。

5. **为开发人员提供安全的默认设置。**在软件开发中，通过为开发人员提供安全的构建块，确保默认路径的安全。例如，考虑到SQL注入漏洞的普遍存在会造成实际损害，确保开发人员使用维护良好的库来防止这类漏洞。这种实践也被称为“铺好的道路”或“照亮的路径”，可确保速度和安全性，并减少人为错误。

6. **培养理解安全性的软件开发员工队伍。**通过对软件开发人员进行安全编码最佳实践培训，确保他们理解安全性。此外，可通过更新招聘方式来评估安全知识，并与大学、社区学院、编程训练营以及其他教育机构合作，将安全性融入计算机科学和软件开发课程中，从而帮助转变更广泛的劳动力队伍。

[3] NIST SSDF, PO 1.2, 示例2: “规定政策来具体说明组织对软件的安全要求，并在软件开发生命周期（SDLC）的关键时间点检验合规性（例如，通过网关验证软件缺陷类别，对已发布软件中发现的漏洞做出响应）”

7. **测试安全事件管理（SIEM）和安全编排、自动化和响应（SOAR）集成。**除了进行实地测试外，还与热门的SIEM和SOAR提供商以及精选客户合作，了解事件响应团队如何使用日志来调查疑似或实际的安全事件。很少有软件开发人员有应对安全事件的经验，因此他们创建的日志条目可能不如预期那样对响应人员有帮助。通过与SIEM和SOAR技术以及实际的事件响应专业人员合作，开发团队可以创建能够正确和完整描述情况的日志，从而在安全事件发生期间节省时间并减少不确定性。

8. **与零信任架构（ZTA）保持一致。**例如，将产品部署指南与NIST的零信任架构（ZTA）模型和CISA的零信任成熟度模型保持一致。鼓励客户在其环境中采用这些原则。

支持安全性的业务实践

1. **提供日志记录服务，不收取额外费用。**云服务提供商应承诺在不收取额外费用的情况下生成和存储与安全相关的日志。现场的产品同样应该生成与安全相关的日志，且不收取额外费用。此外，产品应默认记录安全事件，因为许多客户可能在安全事件发生后才理解其价值。这些策略可能需要彻底审查应记录哪些安全事件以提高客户对网络安全状态的意识、客户如何配置日志记录、保留哪些时间段的日志、如何保护日志完整性和存储，以及如何分析日志。在某些情况下，审查可能表明需要重构应用程序的日志管理架构，以使其具有可操作性，并以制造商可接受的成本进行。与事件响应（IR）专家合作可以增加日志对现场调查人员有用的可能性。请参阅有关SIEM的部分。

2. **消除隐藏的费用。**公开承诺绝不会对安全或隐私功能或这些功能的集成收费。例如，在更大范围的身份和访问管理（IAM）方面，有一类服务称为单点登录（SSO）服务。一些制造商会收取更多费用来将其系统连接到SSO服务（有时称为身份提供商）。这种“SSO费用”意味着许多中小型组织（SMO）无法实现良好的身份和访问管理，从而阻碍他们获得强大的安全态势。某些服务会收取更多费用来为用户启用MFA。**安全性不应被定价为奢侈品，而应被视为客户的权利。**一些制造商争辩说，很少有客户会要求这些功能，并且它们的维护成本较高。这种说法忽视了很少有客户会打电话投诉或讨价还价的事实，并非所有客户都能够实际理解这些功能的好处，而且所有功能都有一定的维护成本。然而，我们并没有看到许多制造商对可用性或数据完整性额外收费。类似于在事故中挽救生命的安全带、可折叠转向柱和气囊的成本已被包括在内，支持这些关键属性的成本也已经包含在所有客户支付的价格当中。

3. **奉行开放标准。**实施开放标准，特别是在常见的网络和身份协议方面。当开放标准可用时，避免使用专有协议。

4. **提供升级工具。**许多客户不愿意采用产品的最新版本，包括部署更新且更安全的功能，比如安全网络连接。软件制造商可通过提供有助于减少不确定性和风险的工具来提高客户对新升级的采用。作为激励客户的一种方式，可以为客户提供免费许可证，以便在测试环境中测试升级和补丁。

原则2：奉行完全的透明度和问责制

解释

软件制造商应该提供安全可靠的产品，凭借自己的这一能力在制造商社区中与众不同，并对此感到自豪。

让我们来解决有关透明度的一个常见担忧。从业人员在讨论完全的透明度时，对话往往会陷入困境，因为他们担心这是在“为攻击者提供路线图”。然而，有过多的证据表明，攻击者在没有此类路线图的情况下也能够进行攻击，因此对于能够使直接客户、间接客户、供应链以及整个软件行业受益的透明度，这些担忧应该放在次要位置。

透明度有助于行业建立惯例——换句话说，就是界定“好”的标准。这有助于这些惯例随着时间的推移而发生变化，以响应客户需求、威胁参与者的策略变化或经济效益变化，或技术的演变。透明度有助于资源较少的制造商向那些资源更为成熟、更有能力的制造商学习。关于信息共享的对话不应局限于实时威胁指标，还应包括以下要素。

透明度迫使制造商在开发过程的早期就围绕安全性做出决策，并使其成为业务领导、工程师以及负责安全的专业人员持续进行的活动。透明度能将责任置于产品当中。

关于在“透明度”前选择使用形容词“完全”的说明。如今，软件制造商很少公开有关他们如何开发和维护软件，以及他们如何利用随时间推移积累的数据完善其程序的详细信息。在软件行业，很少有制造商提供关于他们如何设计软件的导览。软件制造商很少有机会了解同行组织如何构建他们的SDLC程序，以及这些程序在客户环境中如何抵御真实的攻击者。整个行业将从更多的信息分享中受益，比如关于衡量安全缺陷成本及消除漏洞类别的策略等主题。由于这些常见做法，每个软件制造商都必须学会如何自行处理产品安全问题。如果不对安全功能加收“奢侈品”费用，或许安全和安全性就会成为成本中心，而不是利润中心，而公司能够通过合作和提供透明度来减轻负担，从而受益。

我们希望重点关注那些将实质性加速软件行业发展的策略。我们已经没有余裕来进行机会主义的渐进式改进。如果我们要共同克服智能和适应性强的对手带来的威胁，就必须奉行更高的透明度，虽然这样的透明度水平在目前会让人感到不适，但这将推动行业向前发展。如今，已经有制造商体现了一些设计安全原则。正如威廉·吉布森（William Gibson）所说：“未来已经来临，只是分布不大均匀而已。”**完全的透明度将有助于传播这些信息，并使防御者比我们的对手受益更多。**

透明度不仅可以帮助同行组织将其SDLC发展成熟，还能带来更多益处。潜在客户和投资者可以更多地了解制造商所做的投资和权衡，以及这些投资为客户创造的安全态势。奉行完全透明度的制造商能够为客户提供信息，帮助他们在做购买决策时不仅考虑价格和功能，还要考虑安全性。

尽管许多组织努力保护其供应链和SDLC，但最近有些公司的构建流程却受到了破坏。奉行完全的透明度应该致使公司公开披露这些攻击，并且促使公司为预防和发现未来的攻击做出改进。这种形式的信息分享将有助于其他组织学习经验教训，使其免遭相同的命运。

原则体现

为体现这一原则，软件制造商应采取以下措施：

默认安全实践

1. **发布与安全相关的统计数据和趋势的汇总信息。**示例主题可以包括客户和管理员对MFA的采用情况以及不安全遗留协议的使用情况。
2. **发布补丁统计数据。**详细说明有百分之多少的客户正在使用最新版本的产品，以及您正在采取什么措施使更新更简便、更可靠。
3. **发布有关未使用权限的数据。**发布有关整个客户群中过度权限的汇总信息，以及为减少客户的攻击面而对产品进行的微调和其他更改。这些未使用的权限可能很适合设置管理员警报，类似于安全带提示音。

安全产品开发实践

1. **建立内部安全控制措施。**许多公司已经看到将其数据转移到云服务提供商的好处。现在，这些云服务提供商已成为攻击者的目标。软件即服务（SaaS）提供商应该发布其内部控制措施的统计数据。例如，SaaS提供商应发布关于其内部部署的防钓鱼MFA的统计数据，例如快速身份在线（FIDO）认证。理想情况下，他们应该能够声明没有员工能够在未经过防钓鱼MFA验证的情况下访问客户数据或其他敏感数据。

2. **发布高级威胁模型。**设计安全的产品始于书面威胁模型，这些模型需要描述开发者想要保护什么以及免受谁的攻击。有效的威胁模型应该基于外部入侵的发生方式，并应涵盖企业和开发环境，以及软件制造商希望产品在客户环境中的使用方式。

3. **发布详细的安全软件开发生命周期（SDLC）自我声明。**遵循NIST的SSDF或其他类似框架的制造商正在积极努力实现成熟的软件开发生命周期。制造商发布自我声明，说明他们已实施哪些控制措施以及针对的是哪些产品，将能表明他们遵循这些最佳实践的承诺，并增强客户的信心。例如，其他证明方案包括以色列网络供应链方法论（Israel Cyber Supply Chain Methodology）。

4. **奉行漏洞透明度。**发布一条承诺，确保已识别的产品漏洞将作为正确且完整的常见漏洞和曝光（CVE）条目进行发布。对于识别漏洞根本原因的常见弱点枚举字段尤其如此。公共CVE数据库越正确和完整，行业就越能追踪产品如何变得更安全，以及哪些类别的漏洞最为普遍。然而，要警惕将CVE视为负面指标的诱惑，因为这些数字也是健康的代码分析和测试社区的标志。随着制造商实施设计安全理念，开始时他们的原始CVE计数可能会增加，这是因为他们会对现有代码中的漏洞进行更全面的发现和修复。制造商应发布对过去漏洞的分析，包括为解决整个类别的漏洞所采取的任何方式和措施。例如，如果一家公司的大部分CVE与跨站脚本攻击（XSS）有关，则应记录根本原因分析、响应方式（例如转向防止XSS的Web模板框架）和结果，这将向客户发出信号，表明他们不会成为这类漏洞的受害者，因为人们对其缓解措施的了解已经有几十年时间。

5. **发布软件物料清单（SBOM）。**制造商应该掌控他们的供应链。组织应为每个产品构建和维护软件物料清单（SBOM）[2]，向供应商请求数据，并向下游客户和用户提供SBOM。这将帮助组织表明他们努力了解产品创建过程中使用的组件、他们应对新识别的风险的能力，同时还可以帮助客户了解在供应链中的某个模块出现新发现的漏洞时该如何应对。作为参考，日本经济产业省（METI）

已发布《针对软件管理的软件物料清单（SBOM）引入指南》透明度应该扩展到嵌入式设备中的固件以及AI/机器学习（ML）中使用的数据和模型。除了在购买决策和运营能力方面提供帮助之外，SBOM在检测和应对恶意供应链攻击的基础设施中也发挥着重要作用。

6. **发布漏洞披露政策。**发布漏洞披露政策，该政策（1）授权对制造商提供的所有产品进行测试以及这些测试的条件，（2）为符合该政策的行动提供法律安全港，并且（3）允许在设定的时间期限后公开披露漏洞。制造商应对发现的漏洞进行根本原因分析，并在尽可能的范围内采取措施消除整个漏洞类别。有关参考语言，请参阅CISA的漏洞披露政策模板。

支持安全性的业务实践

1. **公开任命一位负责设计安全的高管作为发起人。**在许多组织中，安全性（如质量）被分派给技术团队，而这些团队进行结构性更改以大幅提升产品安全性的能力有限。公开任命一名企业高管来监督设计安全计划，能够把产品的安全性转变为最高级别的业务关注点。

2. **发布设计安全路线图。**制造商应记录对其软件开发生命周期（SDLC）所做的改变，以提高客户安全性，包括有关实地测试报告的详细信息、为消除整个类别的漏洞所采取的行动，以及其他原则中列出的其他项目。与质量改进工作类似，安全改进计划也有明确的规划、控制和改进阶段。本着“以实际行动示人，而非空谈”的精神，发布这些阶段背后的路线图和细节将建立人们对产品设计安全的信心。在取得有意义的进展后，制造商可以在透明度报告中详细说明这些进展。通过这样做，制造商不仅表明了他们对设计安全原则的承诺，而且还向他人展示了实际例证，以此激励他人采取类似的计划。

3. **发布内存安全路线图。**制造商可以采取措施，通过迁移现有产品并使用内存安全语言构建新产品来消除最大的一类漏洞。虽然这可能并非在所有情况下都可行，但制造商可以考虑使用内存安全语言开发应用程序包装器，而不是重写整个应用程序。这还可以包括制造商如何更新招聘、培训、代码审查和其他内部流程，以及他们帮助开源社区实现相同目标的方式。

4. **发布结果。**在更新软件开发生命周期（SDLC）以体现设计安全的理念时，组织会发现一些快速取得的成功、更需要资源的成功以及一些意外的挫折。通过展示他们内部的成功和障碍，整个行业都可以从这些结果中吸取经验教训。

原则3：由高层领头

解释

虽然整体理念被称为“设计安全”，但对客户安全的激励措施早在产品设计阶段之前就已经开始了。这些激励措施始于业务目标、隐含和明确的目标以及期望的结果。只有当高层领导人员将安全视为业务优先事项，创建内部激励措施，并在整个组织中倡导一种将安全作为设计要求的文化，才能取得最佳结果。

虽然技术主题方面的专业知识对于产品安全至关重要，但这并不是一个可以完全交给技术人员的问题。这是一个必须从高层开始的业务优先事项。

有些人想知道，如果软件制造商正在采纳前两个原则并产生了有意义的成果，那么第三个原则是否还有必要？公司如何确立其愿景、使命、价值观和文化将会对产品产生影响，而这些要素在很大程度上由高层决定。我们在其他已经在安全和质量方面取得显著进步的行业中看到了这一点。著名的质量专家约瑟夫·莫西·朱兰（J.M. Juran）写道：

> ***取得卓越的领导力需要高层管理人员亲自负责质量管理。在那些取得卓越领导力的公司中，高层管理人员亲自指导了这一倡议。据我所知，没有任何例外情况。[3]***

我们认为安全是产品质量的一个子类别。当安全和质量成为业务要务，而不是仅由技术人员负责的技术职能时，组织将能够更快、更高效地响应其客户的安全需求。此外，投入必要的资源以确保软件安全从一开始就成为核心业务优先事项，将降低解决软件缺陷的长期成本，进而降低国家安全风险。

与领导团队实施企业社会责任（CSR）计划的方式相同，越来越多的人意识到企业董事会，包括软件制造商的董事会，应在引导网络安全计划方面发挥更积极的作用。有时使用企业网络责任（CCR）一词来描述这一新兴的理念。

原则体现

为体现这一原则，软件制造商应采取以下措施：

1. **将设计安全计划的详细信息包含在公司财务报告中。**如果制造商是一家上市公司，在每年的年度报告中添加一个专门介绍“设计安全”工作的部分。汽车公司的年度财务报告中通常包含有关驾驶员和乘客安全的部分，包括有关集中式和分布式质量和安全委员会的信息。在财务报告中详细说明“设计安全”计划将表明组织正在将客户安全与公司财务业绩联系起来，而不仅仅是因为这个词在营销材料中很流行而采用它。

2. **定期向董事会提供报告。**首席信息安全官（CISO）向公司董事会的汇报中通常包括有关当前和规划中的安全计划、威胁、疑似和确认的安全事件以及围绕公司安全态势和健康的其他更新信息。除了接收有关企业安全态势的信息之外，董事会还应要求有关产品安全及其对客户安全影响的信息。董事会不应仅仅依赖CISO，而应主要依赖公司管理层的其他成员来降低客户风险。

3. **赋予负责“设计安全”的高管更多权力。**技术团队拥有“高管支持”的组织与那些业务领导使用标准业务流程亲自管理客户安全改进流程的组织之间存在着显著的差异。“高管支持”一词意味着有人必须推销客户安全计划的理念，而不是将其作为高层业务目标。必须赋予这位高管权力，以影响产品投资，从而实现客户安全结果。

4. **创造有意义的内部激励措施。**在注意不要创造不正当的激励措施的同时，调整奖励体系以匹配其他有价值的行为和结果，从而提高客户安全性。从负责“设计安全”的高层管理人员到产品管理、软件开发、支持、销售、法务和其他组织，将客户安全激励措施融入到招聘、晋升、薪水、奖金、股票期权和业务运营的其他常见流程中。例如，在制定软件开发人员的晋升标准时，应考虑提高产品安全性以及正常运行时间、性能和功能改进等其他标准。

5. **创建一个设计安全委员会。**在某些行业中，组织通常会建立一个中央质量委员会，并在关键部门或业务单位中设立质量代表。通过包括集中式和分布式成员，这些团队致力于实现针对高层目标的质量改进，同时接收来自组织深处的遥测数据。同样，设计安全委员会将在整个组织中针对设计安全目标改进安全性。

6. **创建并发展客户委员会。**许多软件制造商都设有客户委员会，其中包括来自不同地区、行业和不同规模的客户。这些委员会可以提供大量关于客户在部署公司产品时取得的成功和面临的挑战等信息。构建委员会议程，专门涵盖解决客户安全问题的主题，即使这并不是参与者目前最关注的问题。考虑客户委员会向哪里汇报以及如何利用参与者的见解来了解产品在部署过程中的安全性。例如，委员会是否偏向于营销和销售目的，还是产品管理？负责“设计安全”的高管应该帮助引导这些客户互动，并将其与本文中的其他要素（如实地研究）联系起来。

设计安全的策略

安全软件开发框架（SSDF），也称为美国国家标准与技术研究院（NIST） SP 800-218，是一套核心的高级安全软件开发实践，可以集成到软件开发生命周期（SDLC）的每个阶段。遵循这些实践可以帮助软件生产商更有效地发现和消除已发布软件中的漏洞，减轻漏洞受攻击的潜在影响，并解决漏洞存在的根本原因以防止未来再次发生。

编写组织鼓励使用设计安全策略，包括参考SSDF实践的原则。软件制造商应制定书面路线图，以在其产品组合中采用更多设计安全软件开发实践。以下是一份路线图最佳实践的举例清单，该清单并不详尽：

- **内存安全编程语言（SSDF PW.6.1）。**尽可能优先使用内存安全语言。编写组织承认，内存特定的缓解措施可能是对遗留代码库有帮助的短期策略。示例包括C/C++语言的改进、硬件缓解措施、地址空间布局随机化（ASLR）、控制流完整性（CFI）以及模糊测试（fuzzing）。尽管如此，越来越多的人一致认为采用内存安全编程语言可以消除这类缺陷，软件制造商应探索采用它们的方式。现代内存安全语言的一些示例包括C#、Rust、Ruby、Java、Go和Swift。阅读NSA的内存安全信息单以了解更多信息。
- **安全硬件基础。**整合能够实现细粒度内存保护的架构功能，例如可扩展传统硬件指令集架构（ISA）的功能硬件增强的RISC指令（CHERI）所描述的架构功能，以及可信平台模块和硬件安全模块等其他功能。如需了解更多信息，请访问剑桥大学的CHERI网页。
- **安全软件组件（SSDF PW 4.1）。**从经过验证的商业、开源和其他第三方开发人员处获取和维护安全可靠的软件组件（例如，软件库、模块、中间件、框架），以确保消费者软件产品的稳健安全性。
- **Web模板框架（SSDF PW.5.1）。**使用实现了自动转义用户输入的Web模板框架，以避免Web攻击，如跨站点脚本攻击。
- **参数化查询（SSDF PW 5.1）。**使用参数化查询，而不是在查询中包含用户输入，以避免SQL注入攻击。
- **静态和动态应用程序安全测试（SAST/DAST）**（SSDF PW.7.2、PW.8.2）。使用这些工具来分析产品源代码和应用程序行为，以检测容易出错的做法。这些工具涵盖了从内存管理不当到容易出错的数据库查询构造（例如，未转义的用户输入导致SQL注入）的问题。SAST和DAST工具可以整合到开发过程中，并作为软件开发的一部分自动运行。SAST和DAST应该补充其他类型的测试，例如单元测试和集成测试，以确保产品符合预期的安全要求。发现问题后，制造商应进行根本原因分析，以系统地解决漏洞。

- **代码审查**（SSDF PW.7.1、PW.7.2）。努力确保提交到产品中的代码经过质量控制技术，例如由其他开发人员进行的同行审查或“错误播种”。
- **软件物料清单（SBOM）**（SSDF PS.3.2, PW.4.1）。纳入SBOM[4] 的创建，以提供对进入产品的软件集的可见性。
- **漏洞披露计划**（SSDF RV.1.3）。建立漏洞披露计划，允许安全研究人员报告漏洞并在此过程中获得法律安全保障。作为其中的一部分，供应商应建立流程来确定已发现漏洞的根本原因。此类流程应包括确定采用本文档中的任何设计安全实践（或其他类似实践）是否可以防止引入漏洞。
- **常见漏洞和曝光（CVE）的完整性。** 确保已发布的CVE包括根本原因或常见弱点枚举（CWE），以便在整个行业范围内分析软件安全设计缺陷。虽然确保每个CVE都正确、完整可能需要额外的时间，但它允许不同的实体发现有利于所有制造商和客户的行业趋势。有关管理漏洞的更多信息，请参阅CISA专门针对利益相关者的漏洞分类（SSVC）指南。
- **纵深防御。**设计基础设施，使单个安全控制受损不会导致整个系统受损。例如，确保严格规定用户权限并使用访问控制列表，这可以减少被盗账户的影响。此外，软件沙箱技术可以将漏洞隔离，以限制整个应用程序受损。
- **满足网络安全性能目标（CPG）。**设计满足基本安全实践的产品。CISA的网络安全性能目标概述了组织应该实施的基本的基准网络安全措施。此外，有关加强组织态势的更多方法，请参阅英国网络评估框架，该框架与CISA的CPG有相似之处。如果制造商未能满足CPG——例如不要求所有员工进行防网络钓鱼的多重身份验证（MFA）——那么他们就不能被视为提供设计安全的产品。

编写组织认识到这些变化是组织态势的重大转变。因此，应根据定制的威胁建模、关键性、复杂性和对业务的影响来对引入这些变化进行优先排序。可以为新软件引入这些实践，并逐步扩展以涵盖其他用例和产品。在某些情况下，某种产品的关键性和风险态势可能值得加快采用这些实践。在其他情况下，可以将这些实践引入遗留代码库中并随着时间的推移进行纠正。

[4] 编写组织中的一些组织正在探索获取有关软件供应链安全保证的替代方法。

默认安全的策略

除了采用设计安全的开发实践外，编写组织还建议软件制造商在其产品中优先考虑默认安全的配置。他们应该努力在产品更新时遵循这些实践。例如：

- **消除默认密码。**产品不应带有普遍共享的默认密码。为消除默认密码，编写组织建议产品要求管理员在安装和配置期间设置强密码，或者在产品发货时为每台设备设置一个独特且强大的密码。

- **针对特权用户强制执行多重身份验证(MFA)。**我们观察到，许多企业是由未使用MFA保护其账户的管理员管理部署的。鉴于管理员是高价值目标，产品应该将MFA设为默认启用，而非可选启用。此外，系统应定期提示管理员注册MFA，直到他们在其账户上成功启用。荷兰国家网络安全中心（NCSC）有与CISA类似的指南，如需了解更多信息，请访问他们的成熟身份认证资料单。

- **单点登录（SSO）。**IT应用程序应通过现代开放标准实施单点登录支持。示例包括安全断言标记语言（SAML）或OpenID Connect（OIDC）。此功能应默认提供，无需额外费用。

- **安全日志记录。**为客户提供高质量的审计日志，无需额外费用或额外配置。审计日志对于检测潜在的安全事件并对其进行升级处理至关重要。在调查疑似或已确认的安全事件期间，它们也至关重要。考虑最佳实践，例如提供与安全信息和事件管理系统的轻松集成，这些系统具有使用协调世界时（UTC）、标准时区格式和强大文档技术的应用程序编程接口（API）访问。

- **软件授权配置文件。**软件供应商应提供有关授权配置文件的作用及其指定用例的建议。制造商应包括一个明显的警告，如果客户不使用推荐的授权配置文件，则通知客户会存在更高的风险。例如：医生可以查看所有患者记录，但医疗调度员对安排预约所需的某些信息拥有有限的访问权限。

- **优先考虑前瞻安全性，而非向后兼容性。**尽管向后兼容性会对产品安全造成风险，但产品中仍然经常包含并经常启用向后兼容的遗留功能。优先考虑安全性而非向后兼容性，赋予安全团队权力来移除不安全的功能，即使这意味着会导致重大更改。

- **跟踪并缩小“加固指南”的大小。**缩小产品所包含的“加固指南”的大小，并努力确保随着软件新版本的发布，“加固指南”的大小会随着时间的推移而缩小。将“加固指南”的组件整合为产品的默认配置。编写组织认识到，缩短的加固指南是与现有客户持续合作的结果，需要包括用户体验（UX）在内的许多产品团队的努力。

- **考虑安全设置对用户体验的影响。**每个新设置都会增加最终用户的认知负担，应结合它带来的商业利益进行评估。理想情况下，不应该存在设置；相反，最安全的设置应该默认整合到产品中。当需要配置时，默认选项应该有广泛的安全性以抵御常见威胁。

编写组织承认，这些变化可能会对软件的使用方式产生操作上的影响。因此，为了权衡操作和安全考量，客户的意见至关重要。我们认为，制定书面路线图以及优先考虑将这些想法融入组织最关键的产品中并受到高层管理人员的支持，是转向安全软件开发实践的第一步。虽然客户的意见很重要，但我们观察到客户不愿意或不能采用改进标准（通常是网络协议）的重要案例。对于制造商来说，重要的是要为客户创造有意义的激励措施，来让他们保持最新状态，而不是让他们无限期地保持易受攻击的状态。

加固指南与宽松指南

加固指南可能源于产品安全控制从开发开始就未嵌入产品架构的情况。因此，加固指南也可以成为对手确定和利用不安全功能的路线图。许多组织通常不知道加固指南，因此他们将其设备配置设置置于不安全的态势。应该由一种倒置模型（称为宽松指南）来取代这种加固指南，并向用户解释他们应该进行哪些更改，同时列出由此产生的安全风险。这些指南应该由负责安全的从业人员编写，他们可以用清晰的语言解释利弊权衡，从而增加正确应用这些指南的可能性。

编写组织建议软件制造商转向默认安全的方法并提供“宽松指南”，而不是开发加固指南来列出保护产品的方法。这些指南以通俗易懂的语言解释决策的业务风险，并可以提高组织对恶意网络入侵风险的意识。安全权衡应由客户的高级管理人员决定，以便在安全与其他业务需求之间取得平衡。

对客户的建议

编写组织建议组织让他们的供应软件制造商对其产品的安全结果负责。作为其中的一部分，编写组织建议高层管理人员优先考虑购买设计安全和默认安全产品的重要性。这可以通过制定政策来实现，要求IT部门在购买软件之前评估其安全性，以及授权IT部门在必要时予以反对。应授权IT部门制定采购标准，强调设计安全和默认安全实践（本文档中概述的那些实践以及组织制定的其他实践）的重要性。此外，在采购决策中执行这些标准时，IT部门应该得到高层管理人员的支持。组织在决定接受与特定技术产品相关的风险时，应将这些决定正式记录在案，由高级业务管理人员批准，并定期提交给董事会。

支持组织安全态势的关键企业IT服务，例如企业网络、企业身份和访问管理以及安全运作和响应能力，应被视为关键的业务功能，因此这些服务的资金应与其对组织任务成功的重要性保持一致。组织应制定升级这些功能的计划，以便利用采用设计安全和默认安全实践的制造商。

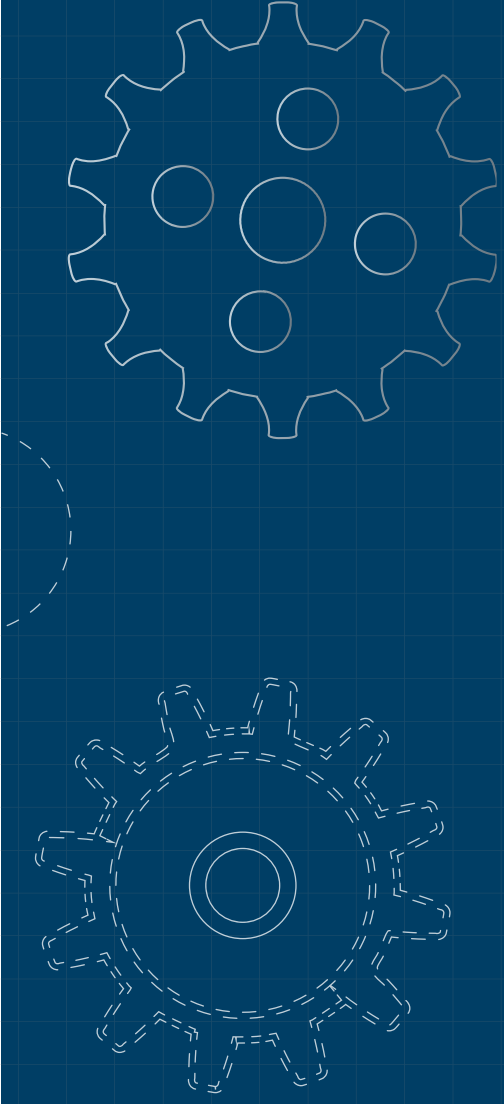
在可能的情况下，组织应努力与其主要IT供应商建立战略关系。这种关系包括组织中多个级别的信任，并提供解决问题和确定共同优先事项的工具。安全应该是这种关系的一个关键要素，组织应该努力在这种关系的正式（例如，合同或供应商协议）和非正式方面强调设计安全和默认安全实践的重要性。组织应该期望其技术供应商对其内部控制态势以及采用设计安全和默认安全实践的路线图保持透明。

除了将默认安全作为组织内的优先事项外，IT领导人员还应与其行业同行合作，以了解哪些产品和服务最能体现这些设计原则。这些领导人员应该协调他们的请求，以帮助制造商优先考虑他们即将推出的安全性举措。通过合作，客户可以帮助向制造商提供有意义的反馈意见，并创造激励措施，使他们优先考虑安全性。

在利用云系统时，组织应确保他们了解与其技术供应商的责任共担模型。即，组织应该清楚供应商的安全责任，而不仅仅是客户的责任。

组织应优先考虑对其安全态势、内部控制以及履行责任共担模型下的义务的能力保持透明的云提供商。

免责声明



资源

CISA

- CISA的SBOM指南
- CISA的跨部门网络安全性能目标
- 技术互操作性指南
- CISA和NIST对软件供应链攻击的防御
- 不安全技术的代价以及我们可以采取的措施 | CISA
- 停止在网络安全问题上推卸责任：为什么公司必须在技术产品中建立安全性 (foreignaffairs.com)
- CISA的专门针对利益相关者的漏洞分类 (SSVC) 指南
- CISA的防网络钓鱼MFA资料单
- 小型企业网络指南 | CISA

NSA

- NSA关于内存安全的网络安全信息单
- NSA的ESF保护软件供应链：供应商最佳实践

FBI

- 理解和应对SolarWinds供应链攻击：联邦视角
- 网络威胁 – 应对和举报
- FBI的网络战略

美国国家标准与技术研究院 (NIST)

- NIST的数字身份指南
- NIST的网络安全框架
- NIST的安全软件开发框架 (SSDF)

澳大利亚网络安全中心 (ACSC)

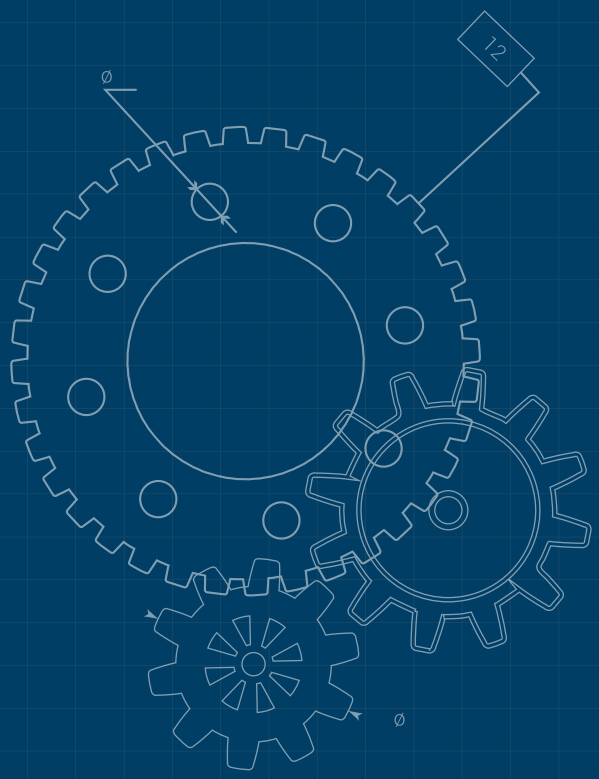
- ACSC的制造商物联网 (IoT) 行为准则指南

英国国家网络安全中心 (UK)

- 英国的网络评估框架
- 英国NCSC的安全开发和部署指南
- 英国NCSC的漏洞管理指南
- 英国NCSC的漏洞披露工具包
- 剑桥大学的CHERI
- 《再见，感谢所有的一切》- NCSC.GOV.UK

加拿大网络安全中心 (CCCS)

- CCCS防范软件供应链攻击指南
- 网络供应链：一种评估风险的方法
- 加拿大网络安全中心的CONTI勒索软件指南



德国联邦信息安全办公室（BSI）

» BSI 基准保护纲要（模块 CON.8）
» 国际标准IEC 62443，第4-1部分
» 2022年德国IT安全状况报告
» BSI的Web应用程序安全实践

荷兰国家网络安全中心

» NCSC-荷兰成熟身份认证资料单

日本国家网络安全事件准备和战略中心（NISC）

» 日本国家网络安全战略

日本经济产业省（METI）

» 针对软件管理的软件物料清单（SBOM）引入指南
» 关于利用OSS并确保其安全性的管理方法的用例示例集

新加坡网络安全局

» 关于安全API开发的技术咨询
» CSA SingCERT漏洞披露政策
» CSA SingCERT事件响应检查清单
» CSA SingCERT 事件响应操作手册
» CSA设计安全框架
» CSA设计安全框架检查清单
» CSA网络威胁建模指南
» CSA 网络安全标签计划

其它

» 复杂系统如何发生故障
» 复杂系统故障的新视角

参考文献

[1] https://csrc.nist.rip/publications/history/ande72.pdf
[2] https://www.cisa.gov/sbom 和 TR 03183-2中有关SBOM的参考资料 https://www.bsi.bund.de/dok/TR-03183
[3] 由约瑟夫·莫西·朱兰（J.M. Juran）撰写的书籍《朱兰-质量源于设计》（Juran on Quality by Design），1992年。