SoK：可持续性是否成为新的可用性？揭穿安全与可持续性根本不相容这一迷思

MaxwellKeleher DavidBarrera加拿大卡尔顿大学
CarletonUniversity,Canada maxwellkeleher@cmail.carleton.ca
david.barrera@carleton.ca

Sonia Chiasson加拿大卡尔顿大学
sonia.chiasson@carleton.ca

arXiv:2603.01958v1 [cs.HC] 2 Mar 2026

摘要

每年，数以百万计仍可正常运行的系统沦为电子废弃物，因为用户在缺乏厂商支持且回收困难的情况下，不得不将系统送往垃圾填埋场。厂商将“网络安全”作为缩短产品支持期的理由，这导致一种普遍却未经审视的看法：网络安全与环境可持续性在根本上相互矛盾；也就是说，要打造安全、耐用且可重复使用的产品，即使不是不可能，也十分困难。为深入了解安全与可持续性之间的复杂关系，我们系统地分析了 29 篇论文，并从中提炼出 155 条可持续性指南，归纳为 12 个可持续性主题。这些主题使我们能够比较可持续人机交互和可持续软件工程的指导原则与网络安全领域的指导原则，并找出其中的一致之处与张力。我们发现，几乎没有证据表明这两个领域之间存在根本性的冲突；少数张力也可以通过审慎考量安全与可持续性目标加以缓解。我们还发现，与可用安全一样，可持续性也面临着“用户是最薄弱环节”这一迷思，以及责任个体化的问题。基于这些相似之处，我们认为，可用安全社区非常适合将可持续性考量纳入其中，因为这两个领域都面临着将责任从个人转向系统性设计的挑战。

1 引言

根据《2024年联合国全球电子废弃物监测报告》， [4], 电子废弃物的产生速度约为收集和回收速度的五倍。供应商对系统支持不佳，人为缩短了系统的使用寿命，并显著加剧了电子废弃物的产生 [61]。导致功能正常的系统被提前淘汰的原因有很多，而网络安全似乎是促使（有时甚至是迫使）用户升级到新系统的重要因素。不过，能否构建出既安全、使用寿命又长，且能让用户在硬件仍可正常使用期间持续再利用并改作他用的系统？这些系统安全、经久耐用，并允许用户在硬件仍可正常运行期间持续再利用并改作他用。

近期的现实案例表明，即便是规模最大的科技公司，在考虑其网络安全决策对环境的影响时，也举步维艰，甚至不愿为之。在2025年3月，谷歌因任由安全证书过期，使数千台原本功能正常的 Chromecast 设备无法使用 [59]。自 Windows 10 于 10 月结束支持以来，微软对 Windows 11 的硬件需求导致数百万台个人电脑无法接收安全更新 2025 [11]。大多数被弃用的 Windows 10 系统仍可正常运行，但用户要么被迫购买新电脑，要么冒着遭受网络攻击的风险。Android设备通常只能获得 1–4 年的安全更新[1]，，之后用户就必须在继续使用不受支持的设备（也就是说，如果发现漏洞，可能没有官方修复来解决该漏洞）和丢弃仍可正常运行的硬件之间做出选择。政策和经济压力或许能促使公司采取行动，但要设计出既安全又经久耐用的系统，这一技术挑战归根结底仍是一个人因问题。它要求我们像可用安全社区长期倡导的那样，在构建安全且易用的系统时，同样重视设计。

我们发现，目前学术界关于可持续性和安全的讨论，反映了早期将安全与可用性置于对立面的论点。Kocksch和Sørensen [53] 声称，可持续性和安全不可能同时得到解决。本质上，“[将可持续性还是安全] 置于关注重点，会 [53]使另一个退居背景”。其他将可持续性与安全联系起来的研究也不加批判地采用了这一假设 [10, 49],，认为存在一个平衡点：更多的安全可能导致更差的可持续性，反之亦然，但这两个领域不可能同时被置于优先地位。迄今文献中提出的可持续安全模型要求分别考量二者

[1]近期，Google Pixel 8/9/10、Sam- sungGalaxy S24/S25 和 Fairphone 等旗舰设备的制造商承诺提供 7 年安全更新 [30, 36, 84],，但这些设备在全球智能手机市场中所占份额仍然很小。

©版权归作者所有 者。允许为个人或课堂用途制作本作品全部或部分内容的数字或纸质副本，无需事先获得许可，前提是不得 ee.
本文版本已提交 可用隐私与安全研讨会（SOUPS）

可持续性和安全。当二者被视为彼此独立的问题时，企业便可以利用安全来为忽视可持续性辩护（正如谷歌、微软和Android的例子所示）。

本文探讨可持续性与网络安全之间是否存在根本性的不相容，这种不相容曾被认为存在于可用性与网络安全之间。如果无法确认这种不相容，那么这两个域之间的关系究竟是什么样的?

贡献。 通过对数字可持续性指南开展系统性文献综述，我们提取出 155 条指南，并将其归纳为 12 个可持续性主题。这些主题涵盖软件开发生命周期，并涉及在整个生命周期中产生影响或受到影响的各类利益相关者。我们将这 12 个可持续性主题与一组 22 条安全设计原则进行比较和对照，以考察可持续性与安全之间存在根本性不相容这一观点。

或许出乎意料的是，我们发现网络安全与可持续性拥有许多共同的设计目标，二者似乎更多是相互契合，而非彼此冲突。我们的主要观察结果是：（1）可持续性与安全设计原则关注许多重叠的目标，以及实现这些目标的策略；（2）潜在的冲突点源于对安全原则的不完整、错误或过度应用；（3）可持续性与安全都需要应对“用户是最薄弱环节”这一迷思。在第 6 节中，我们回顾了三个现实案例，展示如何将这 12 个可持续性主题与安全原则结合使用，以描述系统设计与开发中的安全相关可持续性问题。最后，我们在第 7 节提出一项研究议程，阐述可用安全与可持续性结合的未来研究机会。

2 背景

本节首先对本文使用的关键术语作出定义。

可持续性： 尽量减少电子废弃物和资源浪费造成的环境危害。例如，数字可持续性可以体现为重复使用旧的计算系统，而不是将其作为电子废弃物丢弃。

指南： 指能够体现系统设计与开发最佳实践的设计原则、启发式方法、指导性问题或非功能需求。

系统： 广义上指由硬件和/或软件构成的数字设备或其他计算系统。

寿命终止（EoL）： 指系统不再具备功能，且修理已不可行的状态；然而，系统的组件仍可能被再利用或回收。

2.1 计算机安全设计原则

与可持续性一样，安全也是一个不断演进、持续进行的过程，而非可客观验证的最终状态。因此，安全从业者必须持续评估系统（理想情况下，运用启发式方法、原则和既定的最佳实践），并随着对手自身的演进，不断改进和调整工具与技术。表面上看，安全从业者已经对安全最佳实践形成了直觉性理解，并且存在许多关于安全的原则和启发式方法体系（例如， [24], [65], [69]）。

没有一套原则能够普遍适用，但就本文而言，我们选用了 van Oorschot 的《计算机安全与互联网：从恶意软件到比特币的工具与瑰宝》第一章 [92]中的 22 条计算机安全设计原则，因为它们涵盖面广，且描述直白具体。相比之下，文献中的许多其他原则要么过于宏观，要么针对特定任务，因而不适用于我们的目的。我们认为，van Oorschot 的安全原则在具体性与抽象性之间取得了恰当的平衡。它们涵盖了计算机安全的大多数相关方面（即适用于系统安全、网络安全、可用安全等领域），范围足够广；同时，每条原则的范围又足够明确，可避免冗余和歧义。这些安全原则适用于我们在审阅可持续设计原则时遇到的各种开发与设计情境。表1总结了各项原则，附录 A 则列出了完整定义。

2.2 相关工作

据我们所知，我们的综述涵盖了范围最广的数字可持续性指南。此前的若干可持续性文献综述聚焦于人机交互和设计 [13, 27, 28, 41, 64, 75, 87]。此外，许多先前的文献综述探讨了可持续性与软件工程之间的联系 [3, 6, 26, 31, 32, 39, 58,62, 71, 77, 89, 94]。以往有关数字可持续性的文献综述，重点研究如何利用技术解决环境可持续性问题 [1, 18, 35]。其中一篇过往的文献综述审视了技术的可持续性，具体聚焦于技术的碳排放或能源效率 [25]。我们承认，减少排放和能源消耗是重要目标；然而，我们旨在更全面地理解如何为可持续性而设计。

2.3 可持续性与安全

人们日益开始关注安全决策或实践对可持续性的影响。尽管单凭安全漏洞可能不会促使终端用户丢弃其设备，[49],安全研究人员近期已开始探索通过延长

安全原则	简要说明
HP1：安全内建	从开发周期一开始就将安全融入其中
HP2：面向演进的设计	设计基础架构、机制和协议以支持演进，并定期重新评估安全性 根据不断变化的威胁调整机制
P1：简单性与必要性	尽可能让设计保持简单精简。
P2：安全默认设置	使用安全的默认设置，包括故障安全设计。
P3：开放式设计	邀请并鼓励公开审查和分析。
P4：完全中介	验证每次对每个对象的访问是否具有适当的授权。
P5：隔离舱室	使用强隔离对系统组件进行分区。
P6：最小权限	分配必要的最少权限，并将其持续时间限制在必要的最短时长内。
P7：模块化设计	优先采用能够隔离权限的面向对象设计。
P8：小型可信基	努力缩小必须受信任的组件中的代码规模。
P9：经时间检验的工具	依赖经过时间检验、由专家打造的安全工具和协议。
P10：最少意外	设计符合用户预期的安全机制和用户界面。
P11：用户认同	设计安全机制，让用户愿意采取最安全、最简单的做法。
P12：充足工作因子	调整安全机制，使击败它所需的成本明显超过对手预期投入的资源。
P13：纵深防御	构建相互支援的多层防御。
P14：证据生成	通过事件日志和监控工具记录系统活动。
P15：数据类型验证	验证所有接收到的数据均符合预期属性。
P16：残留清除	会话终止时 , 删除所有敏感数据的痕迹。
P17：信任锚点论证	确保对任何假定信任的基点充满信心，并验证信任假设。
P18：独立确认	通过简单、独立的交叉核查，提高对代码或数据的信心。
P19：请求-响应完整性	验证分布式协议中的响应是否与请求相匹配。
P20：审慎分配	避免向未经身份验证的外部代理分配资源。

表1：van Oorschot 的安全原则摘要 [92]。HP# 表示高层原则，P# 表示原则。完整说明见附录 A。

在设备制造商不再提供安全更新支持后，延长安全更新支持的方式。 [10, 21]。

在一篇发表于 2023年的研讨会论文《将人机交互中的可持续性与网络安全关系视为一个实际问题进行探究 [53],》中，Kocksch和Sørensen讨论了他们通过对信息技术系统开展民族志田野调查所发现的张力。他们进行了一项关于某德国大学数据中心的民族志研究，以及另一项关于丹麦中小型企业信息技术系统的民族志研究。基于这两项民族志研究的观察，Kocksch和Sørensen推断，可持续性与网络安全之间存在根本性的张力，并认为有必要应对这两个议题之间产生的张力。在研究中，他们观察到，当可持续性或网络安全中的一方成为焦点时，另一方就会被置于次要位置。他们指出，要平衡这两个目标，就有必要交替关注可持续性和安全。

2.4 研究方向与范围

本文旨在了解可持续性与安全之间所谓的冲突点究竟源自何处。此外，我们也希望了解这两个领域是否存在相辅相成或协同增效的指导原则，以及这些原则具体体现在哪里。为此，我们对可持续性指南开展了新的整体性综述，涵盖技术的设计与开发，并将其纳入综述范围。与以往的综述不同，我们关注的是如何让数字技术本身更具可持续性，而不是利用数字技术来促进其他领域的可持续性。

3 方法

我们的文献综述采用了引文追踪方法 [23]。我们查阅了给定论文的参考文献（反向检索），以及其他引用该论文的论文（正向检索）。我们使用术语**参考文献**来表示被给定论文引用 的论文（即论文末尾的参考文献列表），并使用术语**引文** 来指引用给定论文的论文。

我们以表6中总结的七篇种子论文为初始集合，开始了引文追踪。我们通过对文献进行初步手工检索 [23] 确定了这些论 [13, 41]文。我们选取了关于可持续人机交互（SHCI） [63, 72] 和可持续软件工程 [8, 22, 57]域的近期综述，以及具有奠基性且被广泛引用的研究成果。Blevis[8] 提出了我们所能找到的最早一组可持续设计指南；他与 Mankoff 等人 [57]的工作似乎标志着 SHCI 研究的起点。尽管其余五篇种子论文本身并未提出可持续性指南，但它们为可持续设计与工程相关的主题或研究领域提供了切入点（例如，

管理产品寿命终止、可持续人机交互和绿色信息技术）。

3.1 论文收集

我们的论文汇编流程总结于图1中。我们使用 Citationchaser [38],这一工具来构建待评审论文列表，该工具会汇编一组论文的参考文献和引文。我们于2025年10月开展了论文检索。

筛选与排除标准。 在评估引文追踪阶段收集的论文时，我们的纳入标准是：论文提出了与可持续性、维修与维护或产品寿命终止相关的指南。我们的排除标准包括以下论文：仅将能源效率或碳排放作为唯一或主要关注点；旨在提升终端用户对可持续性或可持续行为的意识；或聚焦于机器学习或人工智能。

第1轮。 从种子论文开始进行前向与后向搜索的第1轮共获得1119篇论文（148篇引文和971篇被引论文）。在初步审查中，我们考察了每篇论文的标题和摘要，并依据纳入标准和排除标准对论文进行筛选。在摘要审查期间，有873篇论文被排除，并删除了72篇重复项[2]。随后，我们审查了剩余的140篇论文（7篇种子论文和其余133篇收集的论文），以确保这些论文提出了新颖且相关的指南。在完整论文审查期间，有112篇论文被排除，因为它们不包含指南、仅重复其他来源中的指南，或包含与我们的分析不相关的指南（例如，指导教育而非指导设计与开发）。最终，我们得到28篇论文，并对其开展了第一轮主题分析（具体过程详见第3.2节）。

第2轮。 在第1轮中，我们找到了更多文献综述论文，这些论文综述了可能促成新指南的指南和研讨会论文集，尽管这些论文自身并不符合我们的纳入标准。我们将这些文献综述和研讨会论文集用作第2轮论文收集的种子论文（n=40）。第2轮中，这些种子论文共找到1980篇论文（1273篇参考文献和707篇引文）。我们剔除了819篇重复论文（即由不同种子论文产生的论文），随后在摘要审查期间又剔除了1132篇论文。全文审查期间另有17篇被剔除，最终纳入1篇论文进行分析，使纳入论文总数达到29篇。我们将这篇论文纳入主题分析。我们在两轮之后停止检索，因为第2轮仅纳入了1篇论文，且分析过程中没有发现新的编码或主题。

[2]在第一轮中，出现了重复项，原因是在线文献数据库中的索引错误，以及不同的种子论文分别引用了同一篇论文或被同一篇论文引用。

3.2 分析

每轮论文收集结束后，我们从纳入的论文中提取指南文本（以及可用时提供的描述或定义），并将其放入电子表格，作为归纳式主题分析的材料 [12]。我们总共提取了155条指南。我们对提取出的指南进行了开放编码，并在每轮收集结束后审查和合并开放编码，以创建能够描述可持续性指南中常见模式的主题。分析得出了12个可持续性主题（定义见第4节）。这12个主题涵盖了提取出的123条指南。另有32条指南被归入杂项组，因为它们与这12个主题均无关，也无法归纳为一个额外主题。

在分析的第二阶段，我们反复将可持续性主题与 van Oorschot 的 [92] 安全原则（表1、附录 A）进行比较。我们逐一考察每项安全原则与每个可持续性主题之间的关系，并判断该安全原则与主题目标是一致还是存在冲突。例如，可持续性主题可用性 （见第4.2节）显然与安全原则 P10：最少意外和 P11：用户认同相一致，二者都强调符合用户对系统安全的心智模型。避免令人困惑的系统行为尤其重要，因为这可能导致用户弃用一个漏洞本可修复的系统。我们还发现，安全原则 P1：简单性与必要性与这一主题相一致，因为简单的系统更容易让用户理解和学习。最后，我们注意到，安全原则 P20：不情愿分配可能与可用性的目标相冲突，因为要求用户为连接和进程授予权限可能会令他们感到沮丧。

第一作者完成了大部分分析，并与另外两位作者进行了频繁而深入的讨论，由此经过多次迭代来完善结果与见解。当所有作者就主题以及一致与张力之处达成一致后，迭代即告结束。在第4节中，我们描述每个可持续性主题及其与安全设计指南的关系。请注意，表3、表4和表5中标记为一致或张力的单元格可能具有不同程度的重要性；被标记单元格的相对数量并不意味着客观上更为显著的关系。

研究者背景。 研究团队认识到，定性研究方法本质上具有主观性，而进行分析的研究者所具有的独特视角为这项工作增添了价值。大部分分析由第一作者 Maxwell Keleher完成。他是一名计算机科学博士生，专注于人机交互，并拥有软件开发人员和人因研究人员的专业经验。他在学术和职业背景中都积累了丰富的定性分析经验。David：博士

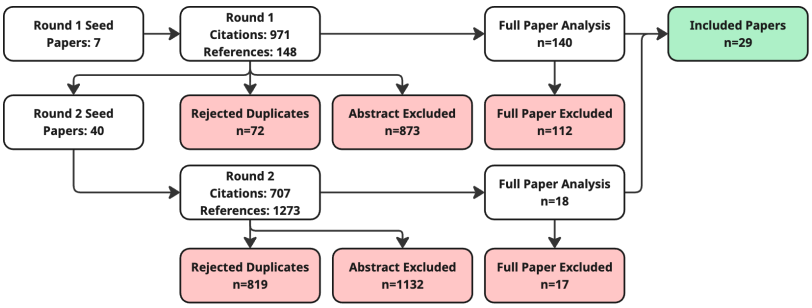

图1：论文收集与筛选流程概览。

David Barrera是计算机科学副教授，研究背景涵盖计算机与网络安全，以及安全与环境可持续性。Sonia：Sonia Chiasson博士是计算机科学教授，研究背景为可用安全与隐私，涵盖定量与定性。

4 文献综述与分析

本节将介绍我们通过对29篇论文中的155条可持续性指南进行主题分析而归纳出的12个可持续性主题（表2）。这些主题对应软件开发生命周期的三个阶段（1：设计与开发，2：使用，3：处置与生命周期终结）。对于每个阶段，我们都会阐述可持续性主题与 van Oorschot 的安全原则之间的一致之处和矛盾之处 [92], ，相关内容汇总于表1。

4.1 设计与开发

我们在文献综述中发现，超过半数的可持续性指南涉及产品生命周期设计与开发阶段的实践。正如在整个设计过程中考虑安全性至关重要，从一开始就考虑可持续性也很有价值。在设计与开发阶段作出的决策会对系统的可持续性和安全性产生重大影响。例如，设计复杂的系统可能会让用户感到困惑，也更难维护，从而对安全性和可持续性均产生负面影响；此外，复杂的系统也可能更难让用户以兼顾安全与隐私的方式进行回收。

接下来，我们逐一定义与设计与开发阶段相关的六个可持续性主题，然后讨论我们关于这些主题如何与van Oorschot的安全设计原则 [92] （见表3）相关联的分析。

简化与减少： 这一主题强调管理可用资源的限制，或者以其他方式尽量减少构建和运行系统所需的资源。设计系统以降低复杂性并尽量减少资源使用，可使其在较长的使用寿命内更易于维护，并减少用户最终弃置系统时产生的电子废弃物。

耐久性与持久性：该主题关注以最大化系统使用寿命的方式设计系统。例如，它鼓励将系统设计得具有韧性且可靠，因为这类系统更有可能长期持续使用。通过考虑其所构建系统的预期使用寿命，设计人员和开发人员可以降低系统被过早弃用的可能性。

兼容性与开放性：该主题强调遵循开放式设计实践，或利用共享资源与基础设施。遵循开放式设计的系统更易于维修与维护，从而延长使用寿命。此外，若原制造商终止支持，这类系统也更有可能切换到第三方基础设施。利用共享基础设施或资源的系统可消除重复工作，并推动基础设施可靠性。

适应与增强： 这一主题涉及设计能够适应不断变化的条件，或能在初始部署后进行增强的系统。适应性强且可增强的系统往往使用寿命更长，并能在用户需求发生变化时降低被弃置的风险。

效率： 这一主题涉及减少设计和开发过程中的浪费，并设计出自身效率更高的系统。其中包括设计系统，使其能够更高效地利用数字资源和物理资源，同时尽量减少电子废弃物和能源消耗。

情境与利益相关者： 要妥善处理可持续性问题，必须在系统的设计与开发过程中恰当地收集情境信息。通过考虑更多不同类型的利益相关者（例如，并非用户但可能受到电子废弃物影响的人），并根据系统预期的使用情境量身设计，系统造成浪费的可能性就会降低。

生命周期阶段	可持续性主题	纳入的论文	# 指南
设计与开发（第4.1节）	简化与减少	[2, 7, 29, 34, 40, 42, 47, 80, 81]	10
	耐久性与持久性	[5, 7, 9, 19, 46, 50, 51, 93]	9
	兼容性与开放性	[5, 7, 9, 45, 47, 50, 51, 67, 68, 73, 76, 81, 91]	16
	适应与增强	[46, 47, 55, 67, 68, 80, 91]	9
	效率	[2, 46, 50, 51, 73, 76]	13
	情境与利益相关者	[2, 5, 9, 14, 19, 40, 70, 81]	19
使用（第4.2节）	用户认同	[7, 8, 34, 47, 80]	10
	可用性	[40, 50, 51, 66, 67, 73, 91]	9
	维修与维护	[7, 20, 50, 51, 52, 55, 73, 91]	13
处置与寿命终结（第 4.3 节）	再利用	[7, 8, 20, 74, 91]	10
	回收	[7, 20]	2
	优雅降级	[7, 20, 46]	3

表 2：从提取的指南和相关论文中归纳出的可持续主题。“# 指南”列表示归入每个主题的指南数量。

	P1	P2	P3	P5	P6	P7	P8	P9	P10	P11	P13	P16	P18	P4	P12	P14	P20	HP2
简化与减少	+				+	+	+					+			-	-	+	
耐久性与持久性	+	+						+			+			-				+
兼容性与开放性			+			+		+					+	-			-	
适应与增强			+	+		+											-	+
效率	+														-			
情境与利益相关者									+	+								

P1：简单性与必要性
P2：安全默认设置
P3：开放式设计
P5：隔离舱室
P6：最小权限
P7：模块化设计
P8：小型可信基
P9：经时间检验的工具
P10：最少意外
P11：用户认同
P13：纵深防御
P16：残留清除
P18：独立确认
P4：完全中介
P12：充足工作因子
P14：证据生成
P20：不情愿分配
HP2：面向演进的设计

表3：矩阵展示了综述中的可持续性主题与 van Oorschot 的安全原则 [92]在设计与开发阶段的关系。绿色“+”单元格表示安全与可持续性指南之间的正向关系。红色“-”单元格表示安全与可持续性指南之间的负向关系或张力。

4.1.1 设计与开发阶段中的协同

我们发现，可持续性与网络安全都倡导并受益于一系列共同的最佳实践：采用简单、分区的设计，遵循开放式设计实践，并注重适应性。这两个域都提醒人们避免采用单体系统，因为这类系统更难维护，而且单个组件发生故障就可能导致整个系统安全失陷。简单的模块化系统浪费更少、更容易保障安全，也能通过更换单个组件而非丢弃整个系统来延长其使用寿命。例如，P7：模块化设计和P5：隔离舱室等安全原则，通过将系统设计分区来实现简洁性（与简化与减少 主题相符），而分区设计也更便于日后修改（与适应与增强主题相符）。

关键协同： 简单、模块化的系统更容易让安全专家维护，并且更有可能拥有较长的使用寿命。

从安全角度来看，开放式设计便于外部审查，从而发现被忽视的漏洞。对于可持续性而言，开放系统鼓励可靠组件的协作式再利用，也更易于修理。两者都受益于采用经受时间考验、公开可用且广为人知的工具，因为这些工具不太可能限制系统寿命。

关键协同： 开放式设计有助于提升系统的可持续性与安全。

4.1.2 设计与开发阶段中的张力

通过我们的分析，我们发现，对某些安全原则作狭义解读，可能会与可持续性主题的目标产生冲突。例如，安全原则 P12：充足工作因子与可持续性主题效率 存在冲突，因为它通过要求付出超出预期攻击者能力的计算开销来实现安全。在安全领域，有时需要利用低效率来阻止针对系统的恶意交互。在评估具有这些特征的系统的可持续性时，重要的是要将其纳入考虑，

	P1	P3	P5	P9	P10	P11	P20	HP2
用户认同					+	+		
可用性	+				+	+	-	
维修与维护		+	+	+			-	+

P1：简单性与必要性
P3：开放式设计
P5：隔离舱室
P9：经时间检验的工具
P10：最少意外
P11：用户认同
P20：不情愿分配
HP2：面向演进的设计

表4：展示综述中的可持续性主题与 van Oorschot 的安全原则之间关系的矩阵 [92]，适用于使用阶段。绿色“+”单元格表示安全指南与可持续性指南之间存在正向关系。红色“-”单元格表示安全指南与可持续性指南之间存在负向关系或冲突。

将低效在实际使用中发生的可能性纳入考量。同样，P14：证据生成是浪费的，因为它涉及生成并存储额外数据（以便在可能发生、也可能不发生的安全失陷时进行访问）。对于P12：充足工作因子和P14：证据生成，我们认为可以在安全目标与可持续性目标之间协商折中，并且对于导致低效或浪费的安全决策，应当提供充分的理由。

关键张力： 为了防止系统滥用，偶尔降低效率可能是必要的。

4.2 使用

系统生命周期的使用阶段侧重于终端用户相关的主题与实践。与这些主题相关的可持续性指南通常会指导设计人员和开发人员如何构建符合终端用户期望的系统，并在系统所有者社群中促进可持续行为。

事实上，在网络安全领域，同样需要使设计决策符合终端用户的期望，以实现预期的安全行为和结果。安全通常不是终端用户的首要目标，[85], 可持续性似乎也是如此。因此，设计人员和开发人员必须仔细考虑如何鼓励用户采取他们并非出于内在动机而愿意采取的行为。

我们定义了通过分析归纳出的三个与使用阶段相关的可持续性主题，随后阐述这些主题与 van Oorschot 的安全原则之间的一致之处和张力 [92]（见表4）。

用户认同： 这一主题关注如何让用户参与可持续实践，并鼓励他们长期使用现有系统。各项指南中的一项关键见解是，用户行为和认知会对系统的可持续性产生重大影响。换句话说，如果用户认为系统经久耐用或有价值，他们就会使用更长时间并加以维护。反之，如果用户觉得系统廉价且用完即弃，他们就不会在意。

可用性： 这一主题涵盖了与可用性相关的人机交互理念，包括推广以用户为中心的设计方法，以及让用户参与协同设计。采用这些方法设计的系统更可能直观易用、效率更高，因此也不太可能被过早弃用。

维修与维护： 这一主题鼓励采用能够延长系统寿命的功能，例如使系统易于修理和维护的能力。遵循这些指南的系统易于诊断，并且拥有可靠的维修指南。任何会影响使用寿命的组件都易于修理或更换。以这种方式设计的系统具有更长的使用寿命。

4.2.1 使用阶段的一致性

通过分析，我们发现可持续性与网络安全指南之间存在一个共同点：二者都强调以用户为中心的方法的重要性。我们发现，在这两个域中，用户认同都是必要的，因为终端用户的行为可能会显著促进或阻碍总体可持续性或安全目标的实现。在这两种情况下，可预测性都是促使用户采取期望行为（即安全或可持续性实践）的关键。此外，改善系统的易学性也能同时带来安全性与可持续性方面的益处。

若干安全原则能够促成更具可持续性的设计（参见表4中标有“+”的单元格）。例如，P3：开放式设计和P5：隔离式分区能够使系统更易于诊断和修理。P9：经时间检验的工具有助于创建更完善的文档，并提高组件的可获得性。此外，P1：简洁性与必要性可以降低复杂性，并有助于提高耐用性。最后，遵循P11：用户认同和P10：最少意外，可以减少使用阻力，鼓励用户继续使用其系统，从而延长使用寿命。

关键协同： 可用性是鼓励安全且可持续用户行为的关键。

4.2.2 使用阶段的张力

在使用阶段，P20：不情愿分配是唯一存在冲突的安全原则（见表4中的“-”单元格）。严格遵循这一原则可能会增加用户的使用阻力，因为他们必须不断批准共享资源或权限；还可能促使用户因觉得系统使用起来令人沮丧而将其弃置。

	P1	P2	P5	P7	P16	P14	P20	HP2
再利用			+	+	+	-	-	
回收	+				+	-		
优雅降级		+			+	-		+

P1：简单性与必要性
P2：安全默认设置
P5：隔离舱室
P7：模块化设计
P16：残留清除
P14：证据生成
P20：不情愿分配
HP2：面向演进的设计

表5：展示文献综述中的可持续性主题与 van Oorschot 的安全原则之间关系的矩阵 [92]，适用于处置与寿命终止阶段。绿色“+”单元格表示安全指南与可持续性指南之间的正向关系。红色“-”单元格表示安全指南与可持续性指南之间的负向关系或张力。

修理系统的能力是实现可持续性时必须考虑的关键因素，但也可能带来意想不到的安全风险。维修过程中集成被入侵组件或模块可能会引发问题，导致系统被处置。例如，用带有过时固件的组件替换原有组件，可能会引入漏洞、破坏系统，并导致过早处置。或者，有人可能安装恶意硬件来监视系统所有者。已有记录显示，维修技术人员会窥探客户送修的系统 [48]，这可能会让用户对选择修理心存顾虑。同样，易于修理或改装的系统可能会为技术促成的滥用行为提供可乘之机[3]。核实系统维修的合法性（无论是自己还是他人进行的维修）对于确保安全至关重要。可持续性与安全领域的专家必须携手合作，确保可维修性不会引入不必要的安全风险。

关键张力： 恶意行为者可能利用易用维修。

4.3 处置与寿命终结

数字系统的生命周期不可避免地会出现性能退化。虽然系统所有者通常会决定如何处置系统（例如再利用、回收或送往垃圾填埋场），但他们能否轻松地做出可持续的选择，往往取决于设计人员或开发人员在上游做出的决策。此外，所有者可用的公共基础设施存在局限，也可能影响终端用户做出可持续选择的能力，因此，公共服务提供者以及市政基础设施的管理者也可能是相关的利益相关者。我们定义了在分析中归纳出的三个与处置和寿命终结阶段相关的可持续性主题，然后阐述这些主题与 van Oorschot [92] 提出的安全原则之间的一致性和张力（见表5）。

再利用： 此主题涉及系统在初始使用阶段之后的再利用。重复使用旧组件，或设计成便于再利用组件的系统，可避免仍可使用的组件成为电子废弃物。此外，即使原所有者不再使用这些系统，只要其设计便于所有权转移，它们也能拥有较长的使用寿命。

回收： 回收是指拆解系统，并将其组件或材料加工成新材料。设计成易于回收的系统，或在设计中使用回收材料的系统，可最大限度地减少电子废弃物的产生。

优雅降级： 所有系统最终都会达到产品寿命终止阶段，但周密设计系统如何达到并在产品寿命终止阶段运行，可以最大限度地减少对可持续性的负面影响。优雅降级策略的示例包括：自行销毁数据 [46]，可生物降解组件 [20]，逐步弃用功能 [37]，以及迁移到只读状态 [37]。

4.3.1 处置和产品寿命终止阶段的一致性

在分析处置阶段的安全原则和可持续性主题时，我们发现，某些安全原则有助于实现可持续性目标，即尽量减少废弃系统产生的电子废弃物。例如，可持续性主题再利用 鼓励使用现有组件，而安全原则 P5：隔离舱室和 P7：模块化设计则有助于实现这一点。这些安全原则能够促进组件自给自足，从而在组件受损或出现安全漏洞时更容易更换。

关键协同： 分区设计更安全，而且更容易再利用和回收。

总体而言，自动移除不必要数据的做法（安全原则 P16：残留信息清除中对此进行了描述），通过消除敏感数据残留在被再利用或回收的系统上所引发的安全风险，促进了组件在所有三个可持续性主题下实现可持续的产品寿命终止。此外，自动移除数据简化了系统的产品寿命终止流程，因为所有者可能无需手动擦除系统。

关键协同： 自动化数据清理可降低安全风险并 促进可持续处置。

[3]技术助长的虐待（TFA）是一种亲密伴侣暴力，施虐者借助数字技术（例如门铃摄像头、智能追踪器）跟踪或骚扰伴侣，或非法访问伴侣的账户或设备 [82]。

4.3.2 处置与生命周期终结阶段的张力

我们的分析指出，对物理系统及其中数据的处置，可能使可持续性与安全陷入冲突。从物理层面来看，不当处置系统（即未进行回收）会产生电子废弃物。然而，处置系统时若未移除敏感信息，攻击者就可能通过“翻找垃圾”攻击获取数据 [16, 33, 88]。此外，还应考虑系统上信息的数字化处置。系统中数据的累积会降低某些操作的性能；如果攻击者能利用累积的数据实施更有效的攻击，也可能带来安全风险。此外，用户可能不记得系统上以数字形式累积的所有信息，因此处置系统时可能导致其信息泄露。这些风险可能促使人们在处置系统前将其销毁，以避免隐私遭到侵犯。从可持续性的角度来看，这并不可取，因为系统完全可以由其他所有者再利用，也可能含有可回收的宝贵材料。如果只孤立地考虑安全或可持续性中的一方，更有可能造成相应后果；但只要审慎考虑决策对可持续性和安全的双重影响，就能将这些风险降至最低。

关键张力： 若不加审慎地实施，可持续处置策略可能引入安全风险。

4.4 总体关系

尽管表3、表4和表5中的主题与原则呈现出趋于一致的趋势，但若声称安全与可持续性只有一致或冲突两种关系，则过于简单化。在某些情况下，安全或可持续性方面的决策能够使另一方受益；而当决策者考虑到对安全和可持续性的潜在影响时，这种情况最有可能发生。设计人员和开发人员在通过审慎实施与安全失陷作出优先保障安全的决策时，可以减轻可能对可持续性造成的负面影响。下文将进一步阐述我们发现的一致点、冲突点，以及安全原则与可持续性主题之间不存在关联的情形。

安全与可持续性的分离： 三项安全原则——P15：数据类型验证、P17：信任锚点论证和 P19：请求-响应完整性——似乎与可持续性无关。这些原则关注的安全问题比其他安全原则更为细致。与此相对应，细致的可持续性原则也可能与安全无关（例如，关于制造业利用可再生能源的可持续性指南；能源来源不会对网络安全产生显著影响）。

先前研究提出的安全与可持续性必须分别处理这一论断 [10, 49, 53] 并非根本错误，但正如下文所述，它未能体现安全与可持续性之间微妙而复杂的关系。

总体一致性： 表3、表4和表5中大多数非空单元格显示，可持续性主题与安全原则之间存在正向关系。我们认为，可以通过恰当应用可持续性或安全指南，同时提升可持续性和 安全。

具体而言，安全原则 P1：简单性与必要性、P5：隔离舱室和 P7：模块化设计倡导采用能够实现可持续性主题 简化与减少、适应与增强 和维修与维护所述成果的设计实践。在安全领域，简单的分区系统更容易修复漏洞。此外，尽量减少此类系统中恶意行为者的潜在攻击点也有益处，因为敏感信息只需与确有需要的组件共享。在可持续性方面，简单的系统浪费更少，也更易于维护，使用寿命更长。此外，分区系统中的组件更易于更换，而此类系统也更便于通过新增模块或组件进行增强。系统的可维护性既是安全领域的目标，也是可持续性的目标；因此，提高系统的可维护性对两者都有益。

我们发现，在优先考虑用户方面，两者存在显著重合。无论是安全还是可持续性，不必要的阻碍都可能促使用户采取会造成负面影响的行为。例如，令人困惑的安全警报可能导致用户忽视真实风险，或丢弃实际上只需进行软件更新的系统。此外，延长系统使用寿命的关键举措（如维护、修理和再利用）都需要用户认同，因此可用性不佳可能会妨碍系统的耐用性。可用安全策略能够减少安全机制造成的阻碍，进而消除可能导致用户不愿长期使用系统的一个因素。为了鼓励可持续行为，系统设计应契合用户的心智模型；这与可用安全致力于设计安全路径最为简单的系统如出一辙。总体而言，这些契合不仅表明一个领域会对另一个领域产生积极影响，也说明只要从业者在系统设计过程中审慎兼顾这两个维度，安全与可持续性就能相互促进。

总体张力： 我们注意到，三项安全原则——P4：完全中介、P12：充分工作因子和 P20：审慎分配——比其他原则更容易与可持续性主题产生张力。这些原则体现了更为封闭的安全方法。尽管它们描述的做法确实能提升安全性，但应谨慎采用，因为它们可能导致不理想的权衡。事实上，这些原则似乎甚至会与其他安全原则产生张力。例如，要求用户每次

与外部代理交互时都授予权限的封闭系统，很可能会令人沮丧，从而违背 P11：用户认同这一安全原则。这些张力并不表明追求可持续性与安全之间存在根本矛盾，而是反映了现实使用场景中安全问题的复杂性，并表明周密的设计可以减轻对安全或可持续性造成的影响。

P14：证据生成是另一项往往与可持续性主题存在张力的安全原则。在这种情况下，张力源于生成过多或过于详细的日志数据，若在处置期间留在系统上，可能会泄露敏感信息。一种简单的结果可能会鼓励销毁系统，而非回收或再利用，但纳入适当的寿命终止安全功能将更有效地缓解这些风险，并促进可持续的回收或再利用实践。

4.5 局限性

我们的分析受到所选安全原则集合的影响；虽然我们认为这些原则具有代表性，但如果使用其他安全原则开展类似研究，可能会出现其他关系。同样，我们的定性分析依赖于研究团队的主观解读。通过迭代过程和团队成员之间的讨论，我们认为自己的解读是合理的。我们承认其他人可能会识别出不同的个别关系，但相信我们的总体论点成立：安全与可持续性存在许多契合点。最后，我们将论点集中在可持续性与安全的关系上，但其他外部压力（如经济激励）在实践中也可能推动使这两个领域相互对立的决策。

5 个关键要点

安全面临挑战的一个重要原因是，不存在安全已经实现的终点或目标。虽然安全原则为提升系统的安全性提供了宝贵指导，但应用安全原则并不能保证产品或系统绝对安全。实现可持续性也面临类似挑战。我们无法预见系统在现实使用中可能遇到的所有问题，而这些问题可能会缩短其使用寿命。此外，即使采用单项可持续实践，也可能涉及权衡。例如，系统或产品的设计可能超出典型条件下的实际需求；虽然这降低了故障风险，也可能提升耐用性，但同时也意味着在设计与开发中浪费资源。安全领域也存在类似现象：为了避免暴力破解攻击而提高密码复杂度，会增加用户为了记住密码而另寻变通办法的可能性，从而带来新的风险（例如，将密码写下来并放在电脑旁，或在多个账户中重复使用密码）。

对于安全和可持续性而言，都需要凭借经验或情境知识预先筛选指南，只采用与特定用例最相关的指南。设计人员和开发人员在设计与开发过程中应牢记所有指南，并在整个产品生命周期中不断回顾这些指南。

要点 1：用户并非安全或可持续性方面的最薄弱环节。可用安全研究人员一直致力于驳斥“用户是安全方面最薄弱环节”这一迷思（例如，[43]）。这种迷思建立在一种观念之上：用户无法遵循安全最佳实践，是个人的失败，而非指南或系统设计存在系统性问题。Maniates [56] 指出，有关可持续性的类似错误指责也被加诸终端用户身上，并认为监管机构和制造商正在逃避自身责任。处置系统时，用户有责任将其送往垃圾填埋场，或作出更可持续的选择（例如回收系统，或设法再利用）。然而，把责任全归咎于用户，忽视了他们必须在其中采取行动的现实限制。系统是否经过耐久设计？只有在系统仍能正常运行时，才能期待用户继续使用。如果系统无故丧失功能，且用户并未误用，那么系统寿命短暂应归咎于制造商或设计者。系统的设计可能并不支持回收，或者用户居住的地方可能不具备回收电子废弃物的条件。此外，糟糕的数据安全设计可能会妨碍用户在转让系统所有权或处置系统前删除或确认已删除其中的个人数据。事实上，缺少设备数据清除工具，可能会限制用户采取再利用等期望行为。值得庆幸的是，我们在评述中发现的指南通常不会将责任归咎于用户，并且承认设计会影响用户的行为。

要点 2：遵循计算机安全设计原则，更有可能促进而非损害可持续发展目标。如表3、表4和表5所示，遵循安全原则很可能会使产品或系统同时满足若干可持续性主题（或者至少不会对可持续性产生影响，这一点可从大量空白单元格中看出）。以往的研究曾明确或隐含地将可持续性和安全视为两项需要分别处理并加以权衡的问题 [10, 49, 53],，而我们发现，谨慎地应用安全或可持续性指南，能够同时促进这两者。总的来说，侧重流程的安全原则（例如 P3：开放式设计或 P7：模块化设计）往往比更细致或具体的原则（例如 P14：证据生成或 P19：请求—响应完整性）更符合可持续性主题。我们的分析发现，在调查的20项安全原则中，只有4项与可持续发展目标存在（有限的）冲突。

要点 3：教条式地应用指南会加剧安全与可持续性之间的冲突。我们发现，当安全或可持续性被过度优先考虑时，就会出现冲突。对于可能被解读为限制性或集权性的安全原则，这种冲突会更加明显。例如，P4：完全中介原则要求在访问资源之前验证访问权限，而这通常由某种集中式机构来执行。集中授权有助于统一管理和实施安全策略，但也会造成单点故障；如果没有考虑恢复机制，该故障可能导致系统无法使用并产生电子废弃物。周密的设计可以考虑这些故障模式，确保设置回退路径，即使个别组件不可用，也能保留设备功能。

要点 4：与可持续性不同，安全忽视了产品生命周期终结及其之后的阶段。长期以来，人们一直倡导从一开始就将安全纳入数字产品的设计与开发流程 [24, 60],，而不是在系统投产后才“补上安全措施”。然而，在产品生命周期的终点，“设计即安全”这一贯穿设计与开发全过程的理念几乎荡然无存。最安全的删除方式是物理销毁存储介质，但如果存储介质仍有可用的剩余使用寿命，这种做法就不可持续。除了有关数据隐私和删除的基本指导，以及恢复出厂设置选项之外，在产品寿命终止时，安全与可持续性仍未得到协调。一方面，安全决策可能会妨碍与产品寿命终止相关的可持续实践。例如，系统可能被设计为禁止所有权转移，以避免泄露原所有者的数据。另一方面，与系统生命周期终结相关的决策也可能带来安全风险。如果没有实用有效的方法从系统中删除个人数据，用户在转移系统所有权时就会面临个人信息泄露的风险，显然会造成负面的安全后果。或者，如果用户无法可靠地清除自己的数据，可能会选择不回收或再利用系统，从而对可持续性造成负面影响。

6 个安全影响可持续性的现实案例

我们运用可持续性主题和安全原则，分析三个现实案例，展示安全与可持续性在实践中的复杂交互。

Google Chromecast 证书过期：2024年，谷歌停止生产已问世 10 年的第二代 Chromecast [59]。在为该产品提供支持期间，谷歌遵循了适当的安全原则，据我们所知，避免了 Chromecast 系统出现大规模安全问题。然而，谷歌似乎在决定淘汰该产品时，并未制定续期 10 年期根证书存储的计划，导致用户系统在联系谷歌服务器时未能通过证书验证检查，进而出现故障（这是 P2：安全默认设置的另一种体现），实际上使所有受影响的 Chromecast 都变成了电子废弃物。在公众施压后，谷歌发布了一项固件更新，将受影响的系统迁移到新的证书颁发机构，其证书过期日期设为 2045 [90],。但这一修复只是将同样的问题推迟了 20 年，并未解决底层设计缺陷：谷歌没有考虑到系统的密码学栈可能比产品的支持期存续更久。

Microsoft Windows 11： 微软要求 Windows 11 [11] 必须配备 TPM 2.0，从而破坏了与较旧硬件的向后兼容性。TPM 2.0 通过验证启动过程各阶段的完整性，确保仅执行可信的固件和操作系统代码，从而实现 P4：完全中介。然而，微软将这一要求与 Windows 11 绑定，同时结束对 Windows 10 的支持，人为地缩短了数百万台缺少这类硬件的旧款个人电脑的使用寿命。这表明微软未能做到优雅降级：微软没有为仍能提供有意义安全保护的旧硬件提供安全性有所降低但仍可正常使用的安全模型（即不需要 TPM），而是让用户只能二选一。我们注意到，微软在技术上完全有能力允许 Windows 11 在没有 TPM 的情况下运行。诸如 BypassTPMCheck 和 BypassSecureBootCheck 之类的注册表项允许在不受支持的硬件上安装系统，这表明这是一项人为限制，而非技术上无法实现。更具可持续性的做法是，微软为旧硬件提供安全性较低的 Windows 11 配置，并向用户清楚说明其中的安全权衡，而不是迫使他们在使用不受支持的操作系统和丢弃仍可正常工作的硬件之间做出选择。

Android 引导加载程序安全：Android 的锁定引导加载程序对主要设备所有者具有正当的安全用途。它可以阻止加载未经授权的固件，确保验证启动完整性，并防止丢失或被盗设备中的数据遭到提取 [79]。然而，一旦制造商停止提供官方软件支持，锁定的引导加载程序就会阻止设备所有者安装由社区维护的固件，而这类固件本可以延长设备的使用寿命（例如通过 LineageOS [54]）。三星等厂商已从部分设备中移除引导加载程序解锁选项 [83]，；而 Fairphone 等厂商则明确将解锁引导加载程序视为核心功能，并在官方支持结束后仍积极支持定制 Android 操作系统社区。这种差异表明，引导加载程序安全可能被用作迫使硬件过时的工具，而非用于其本来目的：在用户需要保护时保护系统。

7 研究议程

安全与可持续性之间的关系是一个新兴研究领域，仍有许多值得进一步探究的机会。我们的分析表明，人们假定这两个领域彼此不相容，在很大程度上是一种误解；但要破除这一误解，需要开展实证研究、开发新工具，并提供更多可操作指导。鉴于我们发现这一领域与可用安全存在诸多相似之处，尤其是在可用性和责任个体化方面，我们认为可用安全社区完全有能力引领这方面的大部分工作。

1. 安全驱动的电子废弃物实证研究。 我们介绍了几个备受瞩目的案例，说明安全决策导致设备提前弃置，但目前几乎没有关于这一现象普遍程度的实证数据。部分挑战在于，安全更新与功能更新之间的界限往往并不清晰。当用户丢弃不再接收软件更新的设备时，很难确定主要原因究竟是失去安全支持、无法获得新功能、性能下降，还是这些因素共同作用。要理解安全在弃置决策中扮演的角色，需要采用可用安全社区研究用户行为时已经使用的同类实证方法：开展纵向研究、问卷调查和访谈，考察用户在日常生活中如何看待安全与过时问题。

2. 衡量安全与可持续性。 衡量与度量是可持续性和安全领域共同面临的难题。二氧化碳排放是评估环境影响的常用指标，但它无法涵盖可持续性的其他方面，例如系统可修复性或可回收性。我们需要新的指标来揭示安全决策与设备寿命之间的关系。例如，平均弃置时间和平均独立所有者数量或许有助于了解设备或系统在长期持有或再利用方面的设计成效。我们需要开展专项研究，在不同设备类别和使用情境中开发并验证这些指标。

3. 生命周期终结时的安全。我们对安全原则和现实使用案例的分析发现，产品支持期结束时缺乏安全指导（第 5 节，结论 4 ）。尽管隐私社区已关注安全数据删除 [17, 78]，但对于更广泛的生命周期终结挑战，几乎没有指导，例如安全地转移所有权、迁移到社区维护的支持，或负责任地逐步停止安全支持。这些挑战本身也可能带来安全隐患。公开发布固件源代码，能让社区维护者更容易延长设备的使用寿命，但也会让攻击者更容易发现永远无法获得官方修复的漏洞。社区维护的软件可以延长设备使用寿命，但用户必须信任维护者。多年来，可用安全社区一直在应对类似的人因问题，我们希望这些专业知识也能应用于可持续安全领域。

4. 可持续安全分析工具。在安全领域，威胁建模工具（例如攻击树和 STRIDE [86]）可帮助设计人员和开发人员系统地识别潜在攻击向量，以便针对这些向量进行设计。我们没有发现用于推理可持续性威胁的同类工具，尤其是在考虑安全引发的可持续性失效时。开发这类工具（或许可以通过改造现有安全分析方法来实现），有助于设计人员在设计过程的早期预见并缓解可持续性风险。

5. 制定可操作指导。设计指南并不总是为开发人员提供具体、可操作的指导。我们的分析表明，这同时影响可持续性和安全：高级原则适用范围广，但需要投入大量精力才能将其付诸实践，因此可能被忽视。可持续性是一个复杂且依赖具体情境的问题，因此，制定用于理解情境的进程（例如探究性问题、分析工具），可能比提供一刀切指南更有效。鉴于终端用户和开发人员面临的相关挑战相同，让可用安全领域的社群参与这项工作尤为重要。尽管已有一些工作致力于制定监管解决方案 [44], ，但这些方案没有为设计人员或开发人员提供明确的说明。一个有前景的方向是开发可持续安全设计模式：针对常见问题、经过验证且能衔接两个领域的解决方案 [15]。第 6 节中的现实案例表明，也可以定义相应的反模式，以标示常见陷阱。然而，制定此类指导需要安全与可持续性专家合作；若能为从密码学到材料回收等主题建立易于理解的通用语言，将有助于这种合作。

8 结论

我们开展了一项知识系统化研究，以确定网络安全与环境可持续性之间的关系，并从可持续设计文献中的29篇论文归纳出12个可持续性主题。我们将这些主题与一系列广泛的安全设计原则进行了比较，发现遵循安全设计原则更有可能有利于而非损害可持续发展目标。少数存在张力之处，源于对个别原则的狭隘应用，而非不同领域之间存在根本性不相容。与安全领域类似，可持续性也受到“用户是最薄弱环节”这一迷思的影响。总体而言，安全与可持续性之间的一致性不仅表明一个领域会对另一个领域产生积极影响，也说明实践者在系统设计过程中仔细考量这两个维度时，安全与可持续性可以相互促进。我们认为，可用安全社区在推动安全发展且不牺牲可持续性方面具有独特优势。

参考文献

[1] Abdullahi Kutiriko Abubakar、Lee Gillam 和 Nishanth Sastry。《物联网（IoT）在实现联合国（UN）可持续发展目标（SDGs）中的作用：系统综述》。*ACM*计算调查，58(3)，2025年10月。doi: 10.1145/3765516。

[2]Jessica L. Anderson、Steve Chenoweth、Rebecca DeVasher、Richard House、Jessica Livingston、Mark Minster、Corey Taylor、Anneliese Watt 和 Julia M. Williams。《传达可持续性：工程、科学和技术传播课堂中的可持续性与传播》。载于*2008 IEEE*国际专业传播会议，第 1–7 页，2008年7月。DOI：10.1109/IPCC.2008.4610193。

[3] Hina Anwar 和 Dietmar Pfahl。《借助软件分析实现更绿色的软件工程：系统映射》。载于*Euromicro*软件工程与高级应用会议（*SEAA*），第 157–166 页，奥地利维也纳，2017年8月。IEEE。DOI：10.1109/SEAA.2017.56。

[4] Cornelis P. Baldé、Ruediger Kuehr、Tales Yamamoto、Rosie McDonald、Elena D'Angelo、Shahana Althaf、Garam Bel、Otmar Deubzer、Elena Fernandez-Cubillo、Vanessa Forti、Vanessa Gray、Sunil Herat、Shunichi Honda、Giulia Iattoni、Deepali S. Khetriwal、Vittoria Luda di Cortemiglia、Yuliya Lobuntsova、Innocent Nnorom、Noémie Pralat 和 Michelle Wagner。《2024年全球电子废弃物监测》。技术报告 978-92-61-38781-5，联合国训练研究所，2024年11月。

[5] Christoph Becker、Ruzanna Chitchyan、Leticia Duboc、Steve Easterbrook、Birgit Penzenstadler、Norbert Seyff 和 Colin C. Venters。《可持续性设计与软件：卡尔斯克鲁纳宣言》。载于国际软件工程会议，第 467–476 页，意大利佛罗伦萨，2015年5月。IEEE。DOI：10.1109/ICSE.2015. 179。

[6] Kristina Rakneberg Berntsen、Morten Rismo Olsen、Narayan Limbu、An Thien Tran 和 Ricardo Colomo-Palacios。《软件工程中的可持续性——系统映射》。载于 Jezreel Mejia、Mirna Muñoz、Álvaro Rocha、Tomas San Feliu 和 Adriana Peña 编，《软件工程趋势与应用》，第 537 卷，第 23–32 页。Springer International Publishing，Cham，2017年。DOI: 10.1007/978-3-319-48523- 2 3._

[7] Eli Blevis. 推进可持续交互设计：关于材料效应的两种视角。设计哲学论文，4(4):209–230，2006年12月。DOI: 10.2752/144871306X13966268131875。

[8] Eli Blevis。《可持续交互设计：发明与处置、更新与再利用》。载于*SIGCHI*计算机系统人因会议论文集，CHI '07，第 503–512 页，纽约州纽约，美国，2007年。计算机协会。DOI：10. 1145/1240624.1240705。

[9] Eli Blevis、Chris Preist、Daniel Schien 和 Priscilla Ho。《进一步将可持续交互设计与可持续数字基础设施设计相结合》。载于有限条件下计算研讨会，LIMITS '17，第 71–83 页，纽约州纽约，美国，2017年6 月。计算机协会。DOI：10.1145/3080556.3080568。

[10] Conner Bradley 和 David Barrera。《摆脱供应商消亡：延长物联网设备寿命的新范式》。载于*2023*年新安全范式研讨会论文集，NSPW '23，第 1–16 页，纽约州纽约，美国，2023年。计算机协会。DOI: 10.1145/3633500.3633501。

[11]Tom Brank。《什么是 TPM？为什么 Windows 11 需要 TPM？》https: // www.pcmag.com/explainers/what-is-a-tpm-and-why-do-i-need-one-for-windows-11，2025年1月。

[12] VirginiaBraun 和Victoria Clarke。《主题分析：实用指南》。SAGE，2022年。ISBN 978-1-4739- 5324-6。

[13] Christina Bremer、Bran Knowles 和 Adrian Friday。《我们是否承担了太多？：对可持续人机交互领域的批判性综述》。载于*CHI*计算机系统人因会议，CHI '22，纽约州纽约，美国，2022年4月。计算机协会。DOI：10.1145/3491102.3517609。网址：https://doi.org/10.1145/3491102.3517609。

[14] Paul Burger与Marius Christen。迈向可持续性的能力方法。《清洁生产杂志》，19(8):787–795，2011年5月。DOI: 10.1016/j.jclepro. 2010.06.019。

[15] Frank Buschmann、Regine Meunier、Hans Rohnert、Peter Sommerlad与Michael Stal。面向模式的软件架构：模式系统 第*1*卷。Wiley，1996。ISBN 978-0471958697。

[16] Cameron Camp 和 Tony Anscombe。《我如何（本可以）用 100 美元窃取您的企业机密》。白皮书，ESET Research，2023年4月。

[17] Jason Ceci、Hassan Khan、Urs Hengartner 和 Daniel Vogel。《担忧却无效：用户感知、方法以及在处置旧设备时面临的挑战》。发表于可用隐私与安全研讨会（*SOUPS*），第 455–474 页。USENIX 协会，2021年8月。ISBN 978-1-939133-25-0。

[18] Deya Chatterjee 和 Shrisha Rao。《计算可持续性：社会技术视角》。*ACM* 计算问卷，53(5)，2020年9月。DOI：10.1145/ 3409797。

[19] Jay Chen。面向极限感知计算的策略。载于第二届限制内计算研讨会，LIMITS ’16，第 1–6 页，纽约，纽约州，美国，2016年6月。计算机协会。DOI：10.1145/2926676. 2926692。

[20] Tingyu Cheng、Gregory D. Abowd、Hyunjoo Oh 和 Josiah Hester。《瞬态物联网：重新设计电子产品寿命，打造更可持续的联网环境》。载于可持续计算机系统研讨会，HotCarbon ’23，第 1–8 页，纽约，纽约州，美国，2023年8月。计算机协会。DOI：10.1145/3604930.3605723。

[21] Akosua Cobbina 和 Bernard Asare。通过安全且具韧性的物联网软件更新提升环境可持续性。载于 Krishnaswami Srihari、Mohammad T. Khasawneh、Sangwon Yoon 和 Daehan Won 编，灵活自动化与智能制造：自动化与制造的未来：智能、敏捷性与可持续性，第 311–319 页，Cham，2026 年。Springer Nature Switzerland。DOI：10.1007/978-3-032-07675-5 29。_

[22] Marisa Leavitt Cohn。共生式衰败：老化基础设施中的交织生命历程。载于*ACM*计算机支持的协同工作与社会计算会议，CSCW ’16，第 1511–1523 页，纽约，纽约州，美国，2016 年。计算机协会。DOI：10.1145/2818048.2820077。

[23] Harris Cooper、Larry V Hedges 和 Jeffrey C Valen- tine。研究综合与元分析手册。罗素·塞奇基金会，2019年。DOI：10.7758/ 9781610448864。

[24]网络安全与基础设施安全局。《转变网络安全风险的平衡：设计即安全软件的原则与方法》。https://www.cisa.gov/sites/default/files/2023-10/SecureByDesign_1025_508c.pdf，2023年。

[25] Ornela Danushi、Stefano Forti 和 Jacopo Soldani。《碳高效软件设计与开发：系统性文献综述》。*ACM CompututingSurveys*，57(10)，2025年5月。DOI：10.1145/3728638。

[26] Kareem Abbas Dawood、Khaironi Yatim Sharif、A. A. Zaidan、Abdul Azim Abd Ghani、Hazura Binti Zulzalil 和 B. B. Zaidan。《开源软件（OSS）可用性映射与分析：面向可持续开源软件产品》。*IEEE Access*，7:65913–65933，2019年。DOI：10.1109/ACCESS.2019.2914368。

[27]Vânia Paula De Almeida Neris、Kamila Rios Da Hora Rodrigues 和 Renata Firmino Lima。《可持续性与人机交互各方面的系统综述》。载于 Masaaki Kurosu 编，人机交互：应用与服务，第 8512 卷，第 742–753 页。施普林格国际出版，Cham，2014年。DOI：10.1007/978-3-319-07227-2 71。_

[28] Carl DiSalvo、Phoebe Sengers与Hrönn Brynjarsdóttir。绘制可持续人机交互的图景。收录于*CHI*计算机系统人因会议，CHI ’10，第1975–1984页，美国纽约州纽约，2010年4月。计算机协会。DOI: 10.1145/1753326.1753625。

[29] Zachary Englhardt、Felix Hähnlein、Yuxuan Mei、Tong Lin、Connor Masahiro Sun、Zhihan Zhang、Shwetak Patel、Adriana Schulz与Vikram Iyer。将可持续性纳入电子产品设计：障碍与机遇。收录于*CHI*计算机系统人因会议，CHI ’25，第1–17页，日本横滨，2025年4月。计算机协会。DOI: 10.1145/3706598.3713299。

[30] Fairphone。《软件长寿命：打造长久耐用智能手机的关键要素》。https://shop.fairphone. com/software-longevity，2025年。

[31] Gabriel Alberto García-Mireles。《软件过程改进中的环境可持续性：一项系统映射研究》。载于 Jezreel Mejia、Mirna Muñoz、Álvaro Rocha、Tomas San Feliu 和 Adriana Peña 编，《软件工程趋势与应用》，第 537 卷，第 69–78 页，Cham，2017年。Springer International Publishing。DOI：10.1007/978-3-319-48523-2_7。

[32] Gabriel Alberto García-Mireles、Ma Ángeles Moraga、Félix García、Coral Calero 和 Mario Piattini。《环境可持续性目标与软件产品质量之间的相互作用：一项映射研究》。*Informationand Software Technology*，95：108–129，2018年3月。DOI：10.1016/j.infsof.2017.10.002。

[33] Simson L Garfinkel 和 Abhi Shelat. 对逝去数据的追忆：一项关于磁盘净化实践的研究。*IEEE Security & Privacy*，1(1):17–27,2003年。DOI: 10. 1109/MSECP.2003.1176992.

[34] Silke Gegenbauer 和 Elaine M. Huang. 通过理解个人依恋，启发更长寿命电子产品的设计。载于 交互系统设计会议, DIS '12, 第 635–644 页, 纽约, 纽约州, 美国 , 2012 年 6 月。ACM。DOI: 10.1145/2317956.2318052.

[35] Mathyas Giudici、Pietro Crovari和Franca Garzotto。面向环境可持续性的劝导式对话代理：一项综述。*ACM*计算调查，58（6），2025年12月。DOI：10.1145/3774751。

[36] 谷歌。了解Google Pixel手机会在何时获得软件更新。https://support.google. com/pixelphone/answer/4457705?hl=en, 2025年。访问日期：2026年2月17日。

[37] Sandy J. J. Gould和Sarah Wiseman。应对数字服务关闭。*ACM*人机交互，6(CSCW2):1–25，2022年11月。DOI：10. 1145/3555617。

[38] N. R. Haddaway、M. J. Grainger和C. T. Gray。Citationchaser：一个用于在学术检索中追踪前向和后向引文的R包，2021年。

[39] Waqas Haider、Muhammad Ilyas、Shah Khalid 和 Sikandar Ali。影响众包软件开发中可持续性方面的因素：系统性文献综述。《软件杂志：演化与过程》，36(6)，2024年6月。DOI：10.1002/smr.2630。

[40] Maria Håkansson 和 Phoebe Sengers。超越绿色：简朴生活家庭与信息和通信技术。载于计算机系统中的人因因素会议，CHI '13，第2725–2734页，纽约州纽约，美国，2013年4月。计算机协会。DOI：10.1145/2470654.2481378。

[41]Lon Åke Erni Johannes Hansson、Teresa Cerratto Pargman 和 Daniel Sapiens Pargman。可持续人机交互十年：将可持续人机交互与可持续发展目标相联系。载于计算系统中的人因因素会议，CHI '21，第1–19页，纽约州纽约，美国，2021年5月。计算机协会。DOI：10.1145/3411764.3445069。

[42] Mike Hazas、A. J. Bernheim Brush 和 James Scott。《可持续性并非始于个人》。《Inter- actions》，19(5):14–17，2012年9月。DOI：10.1145/ 2334184.2334189。

[43] Jonas Hielscher、Uta Menges、Simon Parkin、Annette Kluge 和 M. Angela Sasse。"不接受安全所需时间的员工，意识还不够"：首席信息安全官对以人为本的安全的看法。载于*USENIX* 安全研讨会，第 2311–2328 页，阿纳海姆（加利福尼亚州），2023年 8 月。USENIX 协会。ISBN 978-1-939133-37-3。网址：https://www.usenix.org/conference/usenixsecurity23/presentation/hielscher.[44]Stacy Higginbotham。《设计长寿：如何打造长寿命联网产品》。https://advocacy.consumerreports.org/research/longevity-by-design-how-to-build-long-lived-connected-products/，2025年。

[45] Jina Huh、Kevin Nam 和 Nikhil Sharma。《寻觅失落的宝藏：理解二手计算设备的再利用》。发表于计算机系统中的人因因素大会，CHI '10，第1875–1878页，纽约，纽约州，美国，2010年。计算机协会。DOI：10.1145/1753326.1753607。

[46] Ravi Jain 和 John Wullert。《挑战：普适计算系统的环境设计》。发表于移动计算与网络国际会议，MobiCom '02，第263–270页，纽约，纽约州，美国，2002年9月。计算机协会。DOI：10.1145/570645.570678。

[47] Heekyoung Jung、Eli Blevis 和 Erik Stolterman。《数字制品物质性的概念化及其对可持续交互设计的影响》。载于设计与复杂性——*DSR*国际会议，加拿大蒙特利尔，2010年7月。

[48] Stephanie Kampf、Asha Tomlinson 和 Jenny Cowley。《我们发现百思买、Mobile Klinik、Canada Computers 等公司的技术人员窥探我们的个人设备》，2023年10月。网址：https: //www.cbc.ca/news/business/marketplace-tech-repair-snooping-1.7000775。

[49] Maxwell Keleher、David Barrera 和 Sonia Chiasson。《平衡安全与耐用性：模块化物联网基础设施的优势》。载于新安全范式研讨会，NSPW '24，第20–34页，纽约，纽约州，美国，2025年。计算机协会。DOI：10.1145/3703465.3703468。

[50] Amin Khalifeh、Peter Farrell、Mohammad Alrousan、Shaima Alwardat 和 Masar Faisal。将可持续性纳入软件项目：概念框架。《国际商业项目管理期刊》，13(6):1339–1361，2020年7月。DOI：10.1108/ IJMPB-12-2019-0289。

[51]Amin Khalifeh、Ahmad Samed Al-Adwan、Mohammed Kasem Alrousan、Husam Yaseen、Belal Mathani 与 Firas Rashed Wahsheh。探索可持续性与项目成功的关联：面向软件行业的拟议框架。可持续性，15（22），2023年11月。DOI: 10.3390/su152215957.

[52] Damla Kilic 与 Neelima Sailaja。以用户为中心的修复：从当前实践到未来设计。载于 Norbert A. Streitz 与 Shin'ichi Konomi 主编，分布式、环境与普适交互，第14718卷，第52–71页。Springer Nature Switzerland, Cham，2024年。DOI: 10.1007/978-3-031-59988-0 4._

[53] Laura Kocksch 和 Estrid Sørensen。《在人机交互中探讨可持续性与网络安全的交汇这一实践问题：提交至WS27研讨会：气候变化领域的人机交互：构想可持续未来》。载于气候变化领域的人机交互：构想可持续未来：*CHI 2023* 研讨会，2023年4月。网址：https://sites.google.com/fbk.eu/ hci-climate-change。气候变化领域的人机交互：构想可持续未来：CHI 2023研讨会；会议日期：2023年4月28日至2023年4月28日。

[54] LineageOS项目。LineageOS。https://lineageos.org，2025年。访问日期：2026年2月18日。

[55] Jasmine Lu 和 Pedro Lopes。《消除电子废弃物》。《ACM Trans. Comput.-Hum. Interact.》，31(6)，2024年12月。DOI：10.1145/3674505。

[56] Michael Maniates。个体化：种一棵树、买一辆自行车，就能拯救世界吗？，第43–66页。麻省理工学院出版社，2002年6月。DOI: 10.7551/mitpress/2097.003.0005。

[57] Jennifer C. Mankoff、Eli Blevis、Alan Borning、Batya Friedman、Susan R. Fussell、Jay Hasbrouck、Allison Woodruff和Phoebe Sengers。环境可持续性与交互。载于*CHI*计算系统人因会议，CHI EA '07，第2121–2124页，美国纽约州纽约，2007年。计算机协会。DOI: 10.1145/1240866.1240963。

[58] C. Marimuthu和K. Chandrasekaran。绿色与可持续软件的软件工程方面：一项系统映射研究。载于软件工程创新会议，ISEC '17，第34–44页，美国纽约州纽约，2017年2月。计算机协会。DOI: 10.1145/3021460.3021464。

[59] Liv McMahon。Google Chromecast 用户对电视流媒体故障第二天仍未解决感到愤怒。https://www.bbc.com/news/articles/ckgzm1lgv22o，2025年3月。

[60]微软。微软安全开发生命周期。https://www.microsoft.com/en- us/securityengineering/sdl，2023年。访问日期：2026-01-22。

[61]Batoul Modarress Fathi、Alexander Ansari和Al Ansari。物联网通过电子废弃物对环境可持续性的威胁。可持续性，14(16)，2022年8月。DOI: 10.3390/su141610161。

[62] Ana Carolina Moises De Souza、Daniela Soares Cruzes、Letizia Jaccheri和John Krogstie。软件开发的社会可持续性方法：系统性文献综述。载于Regine Kadgien、Andreas Jedlitschka、Andrea Janes、Valentina Lenarduzzi和Xiaozhou Li编，以产品为中心的软件过程改进，第14483卷，第478–494页，柏林，海德堡，2023年。施普林格出版社。DOI: 10.1007/978-3- 031-49266-2 33。_

[63]Brunna C. Mourão、Leila Karita 和 Ivan Do Carmo Machado。《绿色与可持续软件工程——系统映射研究》。载于巴西软件质量研讨会，SBQS '18，第121–130页，纽约州纽约，美国，2018年10月。计算机协会。DOI：10.1145/3275245.3275258。

[64] Ahmed M. Moustafa。审视可持续产品设计技术：对趋势、创新、挑战和未来前景的全面探索。《循环经济与可持续性》，5(5):4421–4445，2025年10月。DOI：10.1007/s43615-025-00591-7。

[65] 美国国家标准与技术研究院。《NIST网络安全框架（CSF）2.0》。技术报告 NIST CSWP 29，美国国家标准与技术研究院，马里兰州盖瑟斯堡，2024年2月。URLhttps://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf。

[66] Marketta Niemelä、Veikko Ikonen、Jaana Leikas、Kristiina Kantola、Minna Kulju、Antti Tammela 和 Mari Ylikauppila。《以人为本的设计：一种以人为本的技术设计方法》。载 Kai Kimppa、Diane Whitehouse、Tiina Kuusela 和 Jackie Phahlamohlaka 编，《信息通信技术与社会》，第78–91页，柏林、海德堡，2014年。施普林格柏林海德堡。DOI：0.1007/978-3-662-44208-1 8。_

[67] Tobias Nyström 和 Moyen Mustaquim。通过包容性创新框架探索可持续人机交互研究维度。载于 David Kreps、Charles Ess、Louise Leenen 和 Kai Kimppa 编，《这一切都将改变——信息通信技术与气候变化：我们能做什么？》，第 537 卷，第 151–165 页。Springer International Publishing，Cham，2018年。DOI: 10.1007/978- 3-319-99605-9 11._

[68] William Odom。《个人清单：迈向持久的人与产品关系》。载于*CHI* 人机交互大会，CHI EA '08，第3777–3782 页，纽约，纽约州，美国，2008。计算机协会。ISBN 9781605580128。DOI: 10.1145/1358628.1358929。

[69] OWASP 基金会。OWASP 十大 Web 应用安全风险。https://owasp.org/www-project- top-ten/，2025年。

[70] Birgit Penzenstadler。《可持续性软件工程工具包——设计虚构》。载于 Aaron Marcus 编，《设计、用户体验与可用性：设计论述》，第 9186 卷，第 634–643 页。施普林格国际出版，Cham，2015年。DOI：10.1007/978-3-319-20886-2_59。

[71] Birgit Penzenstadler、Ankita Raturi、Debra Richardson、Coral Calero、Henning Femmer 和 Xavier Franch。《软件工程可持续性（SE4S）系统映射研究》。载于《软件工程评估与评价》，EASE '14，第 1–14 页，英国伦敦，2014年5月。计算机协会。DOI：10.1145/2601248.2601256。

[72] Birgit Penzenstadler、Ankita Raturi、Debra Richardson 和 Bill Tomlinson。《安全、安保，如今还有可持续性：21世纪的非功能需求》。《IEEE 软件》，31(3)：40–47，2014年。DOI：10.1109/MS.2014.22。

[73] Yen Dieu Pham、Abir Bouraffa 和 Walid Maalej。《ShapeRE：面向可持续软件需求的多维表示》。发表于国际需求工程大会（*RE*），第 358–363 页，瑞士苏黎世，2020 年 8 月。IEEE。DOI：10.1109/RE48521.2020.00048。

[74] James Pierce 和 Eric Paulos。《二手交互：探究围绕家居物品的重新获取与弃置实践》。发表于计算系统中的人为因素大会，CHI '11，第 2385–2394 页，美国纽约州纽约，2011 年。计算机协会。DOI：10.1145/1978942.1979291。

[75] Giacomo Poderi 和 Yvonne Dittrich。《参与式设计与可持续性：PDC 会议论文集文献综述》。发表于参与式设计大会：短论文、情境行动、研讨会和教程，PDC '18，第 1–5 页，美国纽约州纽约，2018 年 8 月。计算机协会。DOI：10.1145/3210604.3210624。

[76] Chris Preist、Daniel Schien 和 Eli Blevis。《理解并减轻设备及云服务设计决策对数字基础设施环境足迹的影响》。载于*CHI*计算机系统人因会议，CHI '16，第 1324–1337 页，2016 年 5 月，美国纽约州纽约。计算机协会。DOI：10.1145/2858036.2858378。

[77] Claudia Raibulet 和 Qiuyang Fu。《面向可持续软件的自适应及其在当前方法中的应用——一项系统文献综述》。载于 Antonia Bertolino，João Pascoal Faria、Patricia Lago 和 Laura Semini 编，信息与通信技术质量，第 2178 卷，第 210–226 页，Cham，2024年。Springer Nature Switzerland。DOI：10.1007/978-3-031- 70245-7 15。_

[78] Joel Reardon、David Basin 和 Srdjan Capkun。《SoK：安全数据删除》。载于*IEEE*安全与隐私研讨会，第 301–315 页，2013 年。DOI：10.1109/SP. 2013.28。

[79] Nilo Redini、Aravind Machiry、Dipanjan Das、Yan-ick Fratantonio、Antonio Bianchi、Eric Gustafson、Yan Shoshitaishvili、Christopher Kruegel 和 Giovanni Vi-gna。《BootStomp：移动设备引导加载程序的安全性》。载于*USENIX* 安全研讨会，第 781–798 页，2017年8月，温哥华，不列颠哥伦比亚省。USENIX 协会。ISBN 978-1-931971-40-9。

[80] Christian Remy、Silke Gegenbauer 和 Elaine M. Huang。《弥合理论与实践差距：将依恋框架应用于可持续人机交互设计的经验与挑战》。发表于*CHI*计算机系统人因会议，CHI '15，第 1305–1314 页，美国纽约州纽约，2015年4月。计算机协会。DOI：10.1145/2702123.2702567。

[81] Christian Remy、Oliver Bates、Vanessa Thomas 和 Elaine M. Huang。《评估可持续性的局限》。发表于计算系统极限研讨会，LIMITS '17，第 103–110 页，美国纽约州纽约，2017年。计算机协会。DOI：10.1145/3080556.3080567。

[82] Michaela M. Rogers、Colleen Fisher、Parveen Ali、Peter Allmark 和 Lisa Fontes。《亲密关系中的技术助长的虐待：范围综述》。《创伤、暴力与虐待》，24(4):2210–2226，2023年10月。DOI：10.1177/15248380221090218。

[83]salvo_giangri。《One UI 8.0》移除了引导加载程序解锁选项。XDA论坛，2025年7月。网址：https://xdaforums.com/t/bootloader-unlocking-option-removed-from-one-ui-8-0.4751904/。访问日期：2026年2月18日。

[84] 三星。安全更新。https://security.samsungmobile.com/workScope.smsb，2025年。

[85] M. Angela Sasse 和 Ivan Flechais。《可用安全：我们为什么需要它？如何实现？》，第 28–47 页。O'Reilly Media, Inc，加利福尼亚州塞瓦斯托波尔，2005 年 1 月。ISBN 9780596008277。

[86] Adam Shostack。《威胁建模：面向安全性的设计》。Wiley，2014年2月。ISBN 978-1118809990。

[87] Lesly Sierra-Fontalvo、Arturo Gonzalez-Quiroga 和 Jaime A. Mesa。深入探讨产品设计中的过时问题：综述。*Heliyon*，9(11)，2023年11月。DOI:10.1016/j.heliyon.2023.e21856。

[88] Iain Sutherland、Gareth Davies、Andy Jones 与 Andrew J. C. Blyth。僵尸硬盘——来自活死人的数据。澳大利亚数字取证会议，伊迪斯·科文大学：156–161，2010年。DOI：10.4225/75/57B2B48D40CE3。

[89] JakubSwacha。可持续软件模型：范围综述。可持续性，14(1)，2022年1月。DOI：10.3390/su14010551。

[90] Iain Thomson。谷歌表示正在为受损的 ChromecastKit 推出修复方案。https://www.theregister. com/2025/03/13/google_chromecast_fix/，2025年3月。访问日期：2026-02-19。

[91] Bill Tomlinson、M. Six Silberman、Donald Patterson、Yue Pan与Eli Blevis。崩溃信息学：增强人机交互中的可持续性与信息通信技术促进发展论述。载于计算机系统中的人因因素会议，CHI '12，第655—-664页，美国纽约州纽约，2012年5月。计算机协会。DOI: 10.1145/2207676.2207770。

[92] Paul C. van Oorschot。计算机安全与互联网：从恶意软件到比特币的工具与瑰宝。信息安全与密码学。Springer国际出版，Cham，2021年。DOI：10.1007/978-3- 030-83411-1。

[93] Ron Wakkary 和 Karen Tanenbaum。《可持续身份：日常设计师的创造力》。载于*CHI*计算机系统人因会议，CHI '09，第 365–374 页，纽约州纽约，美国，2009 年。计算机协会。DOI：10.1145/1518701.1518761。

[94] Nina Wolfram、Patricia Lago 和 Francesco Osborne。《软件工程中的可持续性》。载于可持续互联网与促进可持续发展的信息通信技术（*SustainIT*），第1–7页，葡萄牙丰沙尔，2017年12月。IEEE。DOI：10.23919/SustainIT.2017.8379798。

计算机安全设计原则

以下原则最初刊载于《计算机安全与互联网：从恶意软件到比特币的工具与瑰宝》 [92]的第1章。为方便读者，现经作者和版权所有者许可，转载这些原则（不包括章节交叉引用和作者注释）。

两项高级原则：

HP1：安全内建。 从开发周期的初始设计阶段开始就将安全融入其中，因为安全设计往往需要核心架构的支持；如果安全只是后期附加的功能，就可能缺少这种支持。明确说明安全机制的设计目标 以及它们不 旨在实现的内容，因为不了解目标就无法评估其有效性。在设计与分析文档中，明确说明所有与安全相关的假设，尤其是与信任和可信方相关的假设（支持P17 ）；请注意，安全策略本身可能并未明确规定这些假设。

HP2：面向演进的设计。 设计基础架构、机制和协议时应支持演进，包括 算法敏捷性 ，以便在尽量减少对相关组件影响的情况下，平稳升级密码算法（例如加密、哈希）。尽可能支持自动化的 安全软件更新 。鉴于威胁、技术和架构不断演进，应定期重新评估安全机制的有效性，并随时准备按需更新设计和产品。

二十项原则：

P1：简单性与必要性。 尽可能让设计简单、精简。减少所用组件的数量，只保留必不可少的组件；尽量精简功能，优先选择最精简的安装方式，并禁用未使用的功能。设计上的经济与精简——尤其是核心安全机制的设计——能简化分析，减少错误和疏漏。配置初始部署，使非必要的服务和应用程序默认处于禁用状态（与 P2 相关）。

P2：安全默认设置。使用安全的默认设置（注意：默认设置往往不会更改）。对于访问控制，默认拒绝。将服务设计为 故障安全，此处指服务发生故障时采取“关闭”状态（例如拒绝访问）。在方法上，使用显式纳入方式，通过 允许列表 (许可列表)列出获准实体，并拒绝所有其他实体，而不是通过 拒绝列表 (黑名单)排除实体，因为后者会允许所有未列入其中的实体。

P3：开放式设计。 不要依赖设计保密、攻击者的无知或 隐匿式安全。欢迎并鼓励公开审查与分析。例如，高级加密标准是通过公开审查，从一系列公开候选方案中选出的；如今，人们普遍不鼓励使用未公开的密码算法。在不与此原则相矛盾的前提下，在有利之处利用不可预测性，因为随意公开战术防御细节很少有益（没必要告诉窃贼你要去度假，也没必要张贴房屋平面图）；同时谨防泄露会随秘密值变化的错误消息或时序数据，以免秘密信息因此泄露。

P4：完全中介。每次访问每个对象时，最好在授予访问权限之前立即进行验证，确认主体拥有适当的授权。验证授权需要进行身份认证（核实身份）、检查相关主体是否获得授权，并检查请求的完整性（请求必须在由合法方发出后未被篡改——参见 P19）。

P5：隔离舱室。使用强隔离结构（容器）对系统组件进行隔离，以管理或阻止组件间通信、信息泄露和控制。这可以在发生故障时限制损害，并防止权限提升；P6 和 P7 的动机类似。将经过授权的组件间通信限制在具有明确定义接口、可观测的路径上，以便进行中介、筛查并利用瓶颈点。隔离手段包括：进程隔离和内存隔离、磁盘分区、虚拟化、软件防护、 安全区域、网关和防火墙。

P6：最小权限。为任务分配所需的最少权限，并且仅在必要的最短时间内授予。例如，仅在需要时保留超级用户权限；如之后仍有需要，再降权并重新获取权限。对于普通用户权限足以完成的任务，不要使用 Unix *root* 账户。这样可以减少暴露面，并限制意外情况造成的损害。P6 与 P5 和 P7 相辅相成。

P7：模块化设计。避免采用将完整权限嵌入大型单体组件的单体式设计；应倾向于采用面向对象和更细粒度的设计，将权限（包括地址空间）分散到较小的单元或进程中。P6 更侧重于权限框架的使用，P7 则更侧重于基础架构的设计。

P8：小型可信基。对于必须可信的组件，即大型系统在安全方面高度依赖的组件，应尽可能缩小其代码规模。例如，高保障系统会将关键安全服务集中在一个精简的核心操作系统——微内核——中，其较小的规模有利于高效开展安全分析。

P9：经时间检验的工具。应尽可能依赖经时间检验、由专家打造的安全工具，包括协议、密码学原语和工具包，而不是自行设计和实施。历史表明，即使对专家而言，安全设计与实施也很难做到万无一失；因此，强烈不建议业余人士尝试（不要重新发明一个更弱的轮子）。机制和工具在负载下经受考验的时间越长，人们对它们的信心就越强（有时称为浸泡测试）。

P10：最少意外。设计机制及其用户界面时，应确保其行为符合用户预期。让设计符合用户对保护目标的心智模型，以减少危及安全的用户失误。尤其是在错误不可逆转的情况下（例如，将机密数据或秘密信息发送给外部人员），应针对目标用户的经验进行设计；谨防设计虽适合专家，却会导致普通用户犯错。更简单、更易用（即易用的）机制能减少意外。

P11：用户认同。 设计安全机制时，应让用户愿意使用，而不是试图绕过，并确保用户阻力最小的选择也是安全的。应选择能凸显益处、改善用户体验并尽量减少不便的设计方案。若用户认为某种机制耗时、不便或没有明显好处，就可能不遵守。示例：部分谷歌 Gmail 用户自愿使用两步验证方案，在基础密码之外，还会使用发送到用户手机的一次性验证码。

P12：充足工作因子。 对于成功概率会随投入努力而可预测地增加的可配置安全机制，应调整机制，使攻破它的成本

（工作因子）明显超过预期对手类别所拥有的资源。因此，应采用强度适当的防御措施。示例 1：随机生成的密码学密钥应足够长，以防通过暴力搜索（即穷举猜测攻击）找到。示例 2：禁止使用弱到只需进行可行数量的猜测便会产生不可忽略成功概率的用户自选密码。**P13：纵深防御**。 构建多个相互支援的防御层，迫使攻击者逐一攻破相互独立的层，从而避免出现 单点故障。如果某一层依赖多个可选防御段，则应将各段设计得强度相当（横向防御中的“等高围栏”），横向防御并优先加强最薄弱的部分（聪明的攻击者会跳过最低的栏杆，或攻破 最薄弱环节）。作为一项设计假设，应预料到某些防御措施会因错误而自行失效，也应预料到攻击者攻破其他防御措施的难度会低于预期，甚至会完全绕过它们。

P14：证据生成。 通过事件日志、监控工具及其他手段记录系统活动，以促进问责，帮助理解系统故障并从中恢复，并支持入侵检测工具。例如：稳健的审计跟踪记录支持取证分析工具，以帮助重建与入侵和犯罪活动相关的事件。在许多情况下，促进检测与证据生成的机制可能比直接的攻击防御更具成本效益。

P15：数据类型验证。 验证所有接收到的数据均符合预期（假定）的属性或数据类型。如果预期有数据输入，应确保后续组件不会将其作为代码处理。这可能涉及规范化 数据（例如，分片数据包或编码的 URL 字符）以及更广泛的输入清理（例如，以应对代码注入或命令注入攻击。

P16：残留清除。会话或程序终止时，移除与任务相关的所有敏感数据痕迹，包括密钥，以及可从辅助存储、随机存取存储器和缓存中恢复的任何残留。请注意，常见的文件删除只是移除目录项，而安全删除则旨在确保即使使用取证工具也无法恢复文件内容。与残留清除相关的是，请注意，进程处于活动状态时，信息可能会泄露在别处通过侧信道。

P17：信任锚点论证。确保对任何假定信任的基点所寄予的信心是合理的，或为其提供论证，尤其是在机制以迭代或传递方式从某个基点扩展信任时（例如浏览器中的信任锚点和证书链）。更普遍地说，应尽可能验证信任假设，并在注册、初始化、软件安装，以及软件应用程序、安全密钥或凭证生命周期的起始阶段格外谨慎。**P18：独立确认**。使用简单、独立的交叉核查（例如在本地设备上进行）来提高对代码或数据的信心，尤其是当它们可能来自外部域或通过不可信信道传输时。例如，可通过比较本地计算出的软件应用程序或公钥的密码学哈希值与通过独立信道（语音通话、短信或广受信任的网站）获得的“已知良好”哈希值，来确认其完整性。

P19：请求-响应完整性。 验证响应与请求在名称解析 及其他分布式协议中相匹配。其设计应包含密码完整性检查，以便在给定事务或协议运行中将各步骤相互绑定，从而检测无关或被替换的响应；警惕缺乏身份认证的协议。例如：一项证书请求若指定唯一主体名称或域，则期望响应为该主体的证书；响应证书中的该字段应对照请求进行交叉核对。

P20：不情愿分配。 与未经身份验证的外部代理交互时，应谨慎分配资源或投入精力。对于向所有方开放的服务，应设计机制以减轻蓄意资源消耗。提高发起交互的代理所需付出的精力。（发起电话呼叫的一方不应要求对方回答：你是谁？尽可能进行身份验证。相关提示：进程应谨防被滥用为通道，从而将其权限扩展给未经验证的代理。）

B 分析中使用的论文

Year	作者	标题	引文
2018	Mourão, Brunna C.; Karita, Leila; Do Carmo Machado, Ivan	绿色与可持续软件工程 - 一项系统映射研究	[63]
2021	Hansson, Lon Åke Erni Johannes; Cerratto Pargman, Teresa; Pargman, Daniel Sapiens	可持续人机交互十年：将可持续人机交互与可持续发展目标相连接	[41]
2022	Bremer, Christina; Knowles, Bran; Friday, Adrian	我们是否承担得太多？：对可持续人机交互领域的批判性综述	[13]
2007	Mankoff, Jennifer C.; Blevis, Eli; Borning, Alan; Friedman, Batya; Fussell, Susan R.; Hasbrouck, Jay; Woodruff, Allison; Sengers, Phoebe	环境可持续性与交互	[57]
2007	Blevis, Eli	可持续交互设计：发明与处置、更新与再利用	[8]
2016	Cohn, Marisa Leavitt	《愉悦的衰败：老年基础设施中交织的生命周期》	[22]
2014	Penzenstadler, Birgit; Raturi, Ankita; Richardson, Debra; Tomlinson , Bill	安全、安全，如今还有可持续性：21世纪的非功能需求	[72]

表6：用于前向-后向综述的第1轮种子论文

Year	作者	标题	引文	# 指南
2016	Chen, Jay	面向极限感知计算的策略	[19]	4
2015	Penzenstadler, Birgit	可持续性软件工程工具包——一种设计虚构	[70]	4
2006	Blevis, Eli	推进可持续交互设计：关于材料效应的两种视角	[7]	10
2013	Håkansson, Maria; Sengers, Phoebe	超越绿色：简朴生活家庭与信息与通信技术	[40]	4
2015	Remy, Christian; Gegenbauer, Silke; Huang, Elaine M.	弥合理论与实践差距：应用依恋 框架的经验与挑战：可持续人机交互设计	[80]	7
2002	Jain, Ravi; Wullert, John	挑战：普适计算系统的环境设计	[46]	6
2012	Tomlinson, Bill; Silberman, M. Six; Patterson, Donald; Pan, Yue; Blevis, Eli	崩溃信息学：增强可持续性	[91]	6
2003	Abraham, Martin A.; Nguyen, Nhan	绿色工程：界定原则——桑德斯廷会议的成果。	[2]	9
2010	Jung、Heekyoung; Blevis、Eli; Stolterman、Erik	数字制品的物质性的概念化及其对可持续- 交互设计的启示	[47]	5
2018	Nyström, Tobias; Mustaquim, Moyen	通过包容性创新框架探索可持续人机交互研究维度—— work	[67]	4
2023	Khalifeh, Amin; Al-Adwan, Ahmad Samed; Alrousan, Mohammed Kasem; Yaseen, Husam; Mathani, Belal; Wahsheh, Firas Rashed	探索可持续性与项目成功之间的关系：一个针对.....的拟议框架 软件行业	[51]	8
2010	Huh, Jina; Nam, Kevin; Sharma, Nikhil	寻找失落的宝藏：理解二手计算设备的再利用	[45]	2
2017	Blevis, Eli; Preist, Chris; Schien, Daniel; Ho, Priscilla	进一步将可持续交互设计与可持续数字基础设施相连接 设计	[9]	4
2014	Niemelä, Marketta; Ikonen, Veikko; Leikas, Jaana; Kantola, Kristiina; Kulju, Minna; Tammela, Antti; Ylikauppila, Mari	以人为本的设计：一种以人为本的技术设计方法	[66]	3
2025	Englhardt, Zachary; Hähnlein, Felix; Mei, Yuxuan; Lin, Tong; Sun, Connor Masahiro; Zhang, Zhihan; Patel, Shwetak; Schulz, Adriana; Iyer, Vikram	将可持续性融入电子产品设计：障碍与机遇	[29]	6
2020	Khalifeh, Amin; Farrell, Peter; Alrousan, Mohammad; Alwardat, Shaima; Faisal, Masar	将可持续性融入软件项目：概念框架	[50]	8
2012	Gegenbauer, Silke; Huang, Elaine M.	通过理解个人依恋，启发更长寿命电子产品的设计 ment	[34]	3
2008	Odom, William	个人清单：迈向持久的人-产品关系	[68]	3
2011	Pierce, James; Paulos, Eric	二手交互：调查围绕 家居物品	[74]	4
2020	Pham, Yen Dieu; Bouraffa, Abir; Maalej, Walid	ShapeRE：面向可持续 软件	[73]	9
2012	Hazas, Mike; Brush, A. J. Bernheim; Scott, James	可持续性并非 始于个人	[42]	3
2015	Becker, Christoph; Chitchyan, Ruzanna; Duboc, Leticia; Easterbrook, Steve; Penzenstadler, Birgit; Seyff, Norbert; Venters, Colin C.	可持续性设计与软件：卡尔斯克鲁纳宣言	[5]	9
2007	Blevis, Eli	可持续交互设计：发明与处置、更新与再利用	[8]	5
2017	Remy, Christian; Bates, Oliver; Thomas, Vanessa; Huang, Elaine M.	评估可持续性的局限	[81]	4
2011	Burger, Paul; Christen, Marius	迈向可持续性的能力方法	[14]	4
2023	Cheng, Tingyu; Abowd, Gregory D.; Oh, Hyunjoo; Hester, Josiah	瞬态物联网：重新设计电子产品寿命，打造更可持续的 网络环境	[20]	4
2016	Preist, Chris; Schien, Daniel; Blevis, Eli	理解并减轻设备与云服务设计决策对 数字基础设施环境足迹的影响	[76]	5
2024	Lu, Jasmine; Lopes, Pedro	消除电子废弃物	[55]	3
2024	Kilic, Damla; Sailaja, Neelima	以用户为中心的修复：从当前实践到未来设计	[52]	5

表7：纳入分析的29篇论文。# 指南 列表示每篇论文提取的指南数量。