当 LLM遇见网络安全：一项系统性文献综述

张杰[1]，[2]，卜浩宇[1]，[2]，文慧[1]，陈宇[1]，[2]，李伦[1]，朱洪松[1]
[1] 中国科学院信息工程研究所，中国北京
[2] 中国科学院网络安全学院，中国北京
{zhangjie, buhaoyu, wenhui, chenyu, lilun, zhuhongsong}@iie.ac.cn

摘要

大型语言模型（LLMs）的快速发展开辟了各个领域的新途径，包括网络安全，面临不断演变的威胁格局和对创新技术的需求。 尽管已经开始探索LLMs在网络安全中的应用，但对这一研究领域缺乏全面的概述。 本文通过提供一项系统性文献综述来弥补这一空白，涵盖了180多项作品的分析，涵盖了25个LLMs和超过10个下游场景。 我们全面的概述解决了三个关键研究问题：网络安全导向LLMs的构建，LLMs在各种网络安全任务中的应用，以及这一领域现有的挑战和进一步研究。 本研究旨在揭示LLMs在增强网络安全实践方面的广泛潜力，并作为在该领域应用LLMs的宝贵资源。 我们还维护并定期更新LLMs在网络安全中的实用指南列表，网址为https://github.com/tmylla/Awesome-LLM4Cybersecurity。

关键词网络安全 ·大型语言模型

1 引言

大型语言模型（LLMs），如ChatGPT [1]、Llama [2]及其衍生物 [3, 4, 5] 所代表的突破性模型，标志着人工智能的重大进步。 这些模型利用大量数据集和先进的神经网络架构，展示了在理解和生成人类语言方面的显着能力 [6,7]。 它们不仅为实现人工通用智能（AGI）设立了新的基准，而且在与领域专家合作时表现出独特的适应性和有效性 [8, 9]。 这样的研究使得LLMs可以针对各个领域的特定挑战进行定制，从而促进医疗保健、法律、教育、软件工程等领域的进步和发展 [10,11,12,13,14,15]。 在网络安全领域，探索LLM应用可以为进一步的模型探索和利用奠定基础，同时突出潜在的变革性影响 [16, 17, 18, 19, 20]。

网络安全是一个关键问题，随着不断增长的网络威胁数量，给个人、组织和政府带来重大风险[21, 22, 23]。 现代网络安全的快速发展和动态性质带来了重大挑战，对手不断调整策略以利用漏洞并规避检测[24, 25]。 传统方法如基于签名的检测和基于规则的系统往往难以跟上不断变化的威胁形势，人工智能的进展，特别是大型语言模型，为增强网络安全开辟了新途径[26]。 已经做出努力将大型语言模型应用于网络安全领域。 一方面，开源的大型语言模型（例如LLaMA [2,27]）促进了网络安全增强领域LLMs的发展，如RepairLlama [28]和Hackmentor [29]，解决独特的网络安全挑战。 另一方面，像ChatGPT这样的先进大型语言模型通过提示工程、上下文学习和思维链解决复杂任务，尽管缺乏网络安全特定的训练[30]。 这些初步努力表明大型语言模型在网络安全任务中发挥着积极的作用，取得了令人期待的结果。

表1:LLMs已被应用的主要网络安全任务和应用领域。

	漏洞检测	(不)安全代码生成	程序修复	二进制	IT运营	威胁情报	异常检测	LLM辅助攻击	其他
RQ1	✓	✓	✓	✓	✓	-	-	-	✓
RQ2	✓	✓	✓	-	-	✓	✓	✓	✓
RQ3	-	-	-	-	✓	-	✓	✓	-

尽管LLMs在网络安全领域的初步努力，但该领域面临着一些挑战[17, 31]。 首先，许多研究依赖于案例研究，缺乏全面的方法论，引发了关于可扩展性和可重现性的担忧。 此外，研究领域呈现碎片化，缺乏研究之间的连接和深入分析。 随着LLM研究在该领域中迅速增加，进行系统概述对于引导该领域进入一个新的发展阶段至关重要，其中LLM应用不仅仅是实验性的，而且具有战略影响力[18,19,20]。 因此，本文旨在对网络安全领域定制的领域特定LLMs进行广泛审查，探索LLM在该领域的应用广度，并确定新兴挑战，为未来发展奠定基础。

本调查旨在全面介绍LLM在网络安全中的应用。 我们试图回答三个关键问题：

- RQ1：如何构建面向网络安全领域的LLM?
- RQ2：LLM在网络安全中的潜在应用有哪些?
- RQ3：关于LLM在网络安全应用方面的现有挑战和进一步研究方向是什么?

通过探讨这些问题，我们旨在弥合LLM的进展与其对增强网络安全实践潜力的差距。 我们将深入研究各种网络安全任务和应用，LLM已显示出潜力，如漏洞检测、安全代码生成、程序修复、二进制分析、运营管理、威胁情报、匿名检测和LLM辅助攻击性安全，如表1所示。

对于第一个问题，我们总结了现有网络安全LLM的原则，详细说明了它们的关键技术、用于模型开发的数据以及针对特殊任务训练的领域LLM。 我们提供了构建领域模型的见解，对于希望构建定制LLM以满足特定需求的研究人员和网络安全从业者，如计算限制、私人数据和本地知识库（第3节）。 对于第二个问题，我们对现有LLM在10多个网络安全任务中的使用进行了广泛调查，包括威胁情报、漏洞检测、程序修复等。 这种分析不仅帮助我们了解LLM在各个方面如何有益于网络安全，还使我们能够确定其在应用于领域特定任务时的优势。 通过展示LLM的多样能力，我们旨在澄清它们在增强和转变网络安全领域潜力的可能性（第4节）。 第三个问题强调了在将LLM应用于网络安全时需要克服的挑战。 LLM固有的漏洞和易受攻击性导致这些挑战，尤其是LLM导向的攻击和LLM越狱。 此外，我们还展望了LLM发展的未来方向，为寻求推动该领域发展并利用LLM在网络安全中潜力的研究人员和从业者提供指南（第5节）。

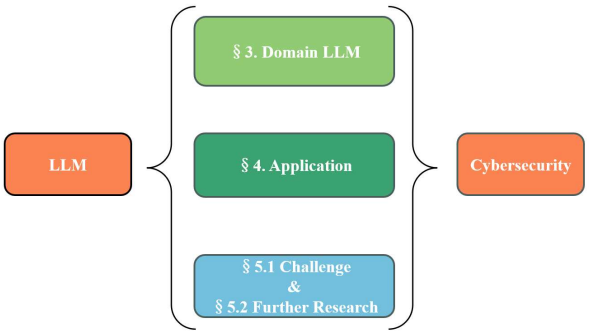


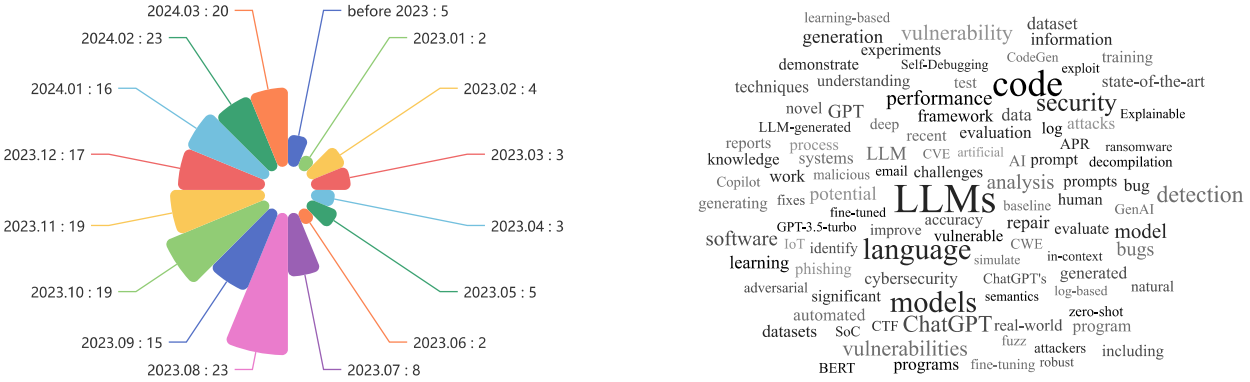
图 1：在网络安全中应用LLM的概念框架。

总之，本文通过对网络安全中最新技术LLM应用的全面审查，突出潜在的益处和挑战，并提出未来研究方向。

本文的后续部分组织如下。 第2节概述了本文的研究范围。 第3节总结了现有的网络安全LLM。第4节详细介绍了LLM在各种网络安全任务中的应用。 第5节强调了未来研究的挑战和有前途的机会。 第6节得出结论。

2初步

在这篇综述论文中，我们系统地调查了LLM在网络安全领域的应用进展，涵盖了自2023年以来的180多篇学术论文。 通过对这些论文进行详尽的研究和全面的分析，我们旨在提供LLM在网络安全中应用的当前状态、挑战和未来方向的详细概述。 如图2所示，这一新兴研究领域一直受到关注，LLM可以用来解决各种领域任务。 这不仅突出了LLM在网络安全中的当前和潜在影响，还为未来研究提供了实用指导。 因此，本节首先总结了从两个方面调查的论文：一个是在网络安全中使用的LLM，另一个是LLM可以应用于的网络安全任务类别。



(a) 按月份的时间分布　　(b) 词云

图2:调查论文的统计数据。

2.1 LLMs在网络安全中的应用

LLMs已经成为人工智能领域中一种具有显着能力的变革性技术，展示出在自然语言理解、生成和推理方面的卓越能力[32, 6, 7]. 这些模型是在大量文本数据上训练的具有潜力在包括网络安全在内的各个领域进行革命性变革[18, 19, 20]。 LLMs在网络安全中的应用有望增强威胁检测、自动化漏洞分析、智能防御机制等方面。

LLMs可以分为两种主要类型：开源和闭源模型。 开源LLMs，如Llama [2]和Mixtral [5]，提供透明度和研究人员定制和微调模型以适应特定网络安全任务的能力。 这种适应性在网络安全场景中尤为重要，例如私人数据和根据定制需求微调模型。 然而，开源LLMs可能缺乏其闭源对手的性能和规模。 另一方面，闭源LLMs，通常被称为商业LLMs，如ChatGPT [1]和Gemini [33]，提供最先进的性能，并由商业实体维护，通常具有受限制的访问权限。 虽然这些模型在准确性和效率方面表现出色，但它们缺乏透明度可能引发对网络安全应用中潜在偏见和限制的担忧。

在网络安全领域，越来越需要能够理解、分析和生成安全代码的智能工具。 基于代码的LLM，如CodeLlama [34]和StarCoder [35,36]，特别适合满足这一需求。 与基于文本的LLM训练于大量自然语言数据不同，基于代码的LLM专门设计用于理解和处理编程语言。 基于代码的LLM训练于跨越多种编程语言的大型代码库，使它们能够捕捉语法、语义和常见编码模式的复杂性。 这种专门的训练使它们能够执行各种任务，包括代码补全、错误检测和自动化代码审查。 在网络安全的背景下，这些能力对于识别潜在漏洞、建议安全编码实践和自动化安全漏洞的修复至关重要。

表2:本文中使用的LLM总结。

组织	LLMs	大小	开源计数		链接
OpenAI	GPT-3.5	175B	×	86	https://chat.openai.com/
	GPT-4	-	×	56	https://chat.openai.com/
	Codex	-	×	13	https://openai.com/blog/openai-codex
	davinci(-002,-003)	175B	×	9	https://openai.com/blog/openai-api
Google	Bard&Gemini	-	×	12	https://gemini.google.com/
	PaLM(-1,-2)	540B	×	7	https://ai.google.dev/models/palm
Anthropic	Claude(-1,-2)	-	×	2	https://claude.ai/
Github	Copilot	-	×	2	https://github.com/features/copilot
Microsoft	BingChat	-	×	2	https://www.bing.com/chat
EleutherAI	GPT-J	6B	✓	2	https://huggingface.co/EleutherAI/gpt-j-6b
	GPT-Neo	2.7B	✓	3	https://huggingface.co/EleutherAI/gpt-neo-2.7B
元	大羊(-1,-2)	7B/13B/70B	✓	38	https://huggingface.co/meta-llama
	LlamaGuard	7B	✓	1	https://huggingface.co/meta-llama/LlamaGuard-7b
	InCoder	1B/6B	✓	4	https://huggingface.co/facebook/incoder-1B
LMSYS	维库纳	7B/13B	✓	12	https://huggingface.co/lmsys/vicuna-7b-v1.5
链家科技	贝尔	7B/13B	✓	1	https://github.com/LianjiaTech/BELLE/
Databricks	多莉	6B	✓	3	https://huggingface.co/databricks/dolly-v1-6b
-	瓜纳科	7B	✓	2	https://huggingface.co/JosephusCheung/Guanaco
Salesforce	CodeGen(-1,-2)	3B/7B/16B	✓	9	https://github.com/salesforce/CodeGen/
	CodeT5	6B	✓	3	https://huggingface.co/Salesforce/codet5p-6b
BigCode	StarCoder(-1,-2)	3B/7B/15B	✓	3	https://huggingface.co/bigcode/
THUDM	ChatGLM	6B	✓	8	https://github.com/THUDM/ChatGLM-6B
KaistAI	普罗米修斯	7B/13B	✓	1	https://github.com/kaistAI/Prometheus
密斯特拉AI	密斯特拉	7B	✓	6	https://huggingface.co/mistralai/Mistral-7B-v0.1
	Mixtral	8*7B	✓	5	https://huggingface.co/mistralai/Mixtral-8x7B-v0.1

2.2 LLMs应用于网络安全的类别

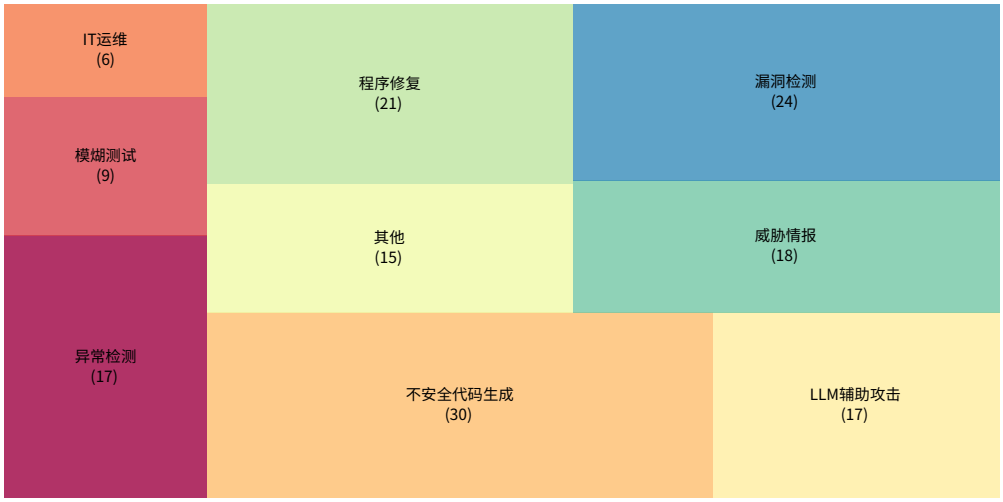


图 3:大型语言模型在网络安全类别上的树状图应用。

随着对互联系统的日益依赖和复杂网络威胁的增加，网络安全已经成为数字时代的一个关键问题[21,23]。 网络安全领域涵盖了一系列旨在保护计算机系统、网络和数据免受未经授权访问、攻击、破坏或干扰的实践、技术和策略[24, 25]。 人工智能技术，特别是大型语言模型，已经显示出在革新网络安全各个方面的巨大潜力[20]。 大型语言模型在网络安全中的应用涵盖了广泛的领域，从威胁检测和分析，安全策略合规性，到自动化漏洞评估和恶意软件分析。

- 漏洞检测：这是网络安全中最重要的任务之一。 结合大型语言模型，已经探索了在检测漏洞方面的新方法。

- （不）安全代码生成：由大型语言模型生成的代码是否存在风险？ 此外，大型语言模型能否通过一些策略纠正它们的代码？
- 程序修复：程序修复是一项任务密集型的工作，修补缺陷需要足够的经验和知识。 许多研究证明了在这个问题上的有效性。
- 二进制：LLMs在处理自然语言和高级编程语言方面表现出色。 验证LLMs理解反汇编的能力也是一个重要的方面。
- IT运营：IT运营涉及许多重复性任务。 LLMs可以被训练来自动化这些任务，提高效率并减少人为错误的可能性。
- 威胁情报：从大量威胁情报文档中提取信息是困难的，一些研究人员已经转向LLM来组织和分析这些庞大而混乱的数据。
- 异常检测：我们主要指的是流量中的恶意流量、系统中的病毒文件、日志中的异常等安全异常。
- LLM辅助攻击：许多人对这些积极应用不满意。 他们发现LLM在发动网络攻击，如钓鱼邮件和渗透测试方面的有效性。
- 其他：除了上述提到的方面，我们还收集了一些研究，证明了LLM在网络安全领域的重要性，尽管他们领域内LLM应用的研究较少。

3 RQ1：如何构建面向网络安全领域的LLM?

网络安全领域面临不断升级的威胁，要求智能和高效的解决方案来对抗复杂和不断演变的攻击[37, 38, 39]。 LLMs为网络安全社区提供了新的机遇[18, 19]。
经过大量数据训练，LLMs已经获得了丰富的知识并发展出强大的理解和推理能力，为网络安全提供了强大的决策支持。

推进网络安全需要定制领域的LLMs，利用它们学习特定领域的数据和知识的潜力。 本节首先关注构建网络安全LLMs的关键技术，包括LLMs的持续预训练（CPT）[40,41]和监督微调（SFT）[42,43]等训练方法，以及使用全参数训练和参数高效微调（PEFT）[44]等技术实现。 然后，我们介绍了几个专为评估LLMs网络安全能力而设计的领域数据集[45,46,47]，这些数据集可以指导在构建网络安全LLMs时选择适合的LLMs作为基础模型。最后，我们总结了通过微调通用LLMs构建网络安全领域特定模型的现有工作[48, 29]，包括但不限于漏洞检测、程序修复、安全代码生成等。

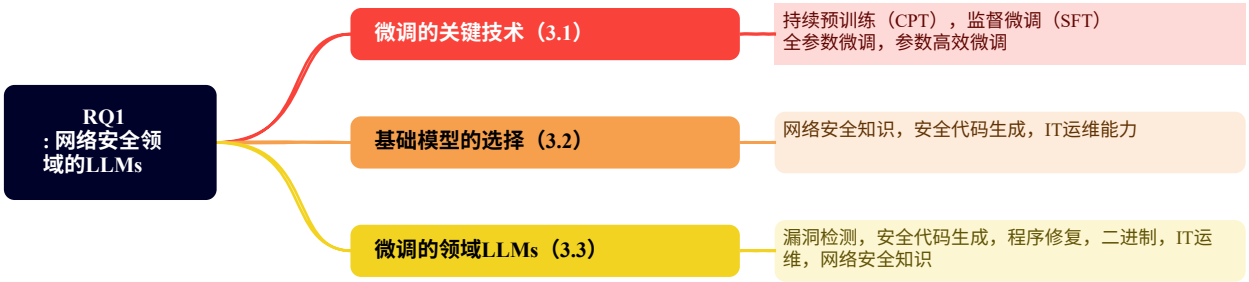


图 4： **RQ1的概述。**

3.1 构建领域LLMs的关键技术

LLMs利用变压器架构和自监督预训练策略[49, 50, 32]，展现出优越的理解和内容生成能力。 然而，从头开始为网络安全开发专门的LLM将需要大量的计算资源，对大多数研究团队来说是不切实际的。 幸运的是，现有的通用LLMs已经获得了广泛的知识，并展现出了显着的泛化能力[2, 27, 51, 5]。 通过将这些预训练的LLMs与网络安全特定数据集集成用于训练目的，我们可以采用更高效的方法来增强模型。 这种方法不仅显着降低了预训练的计算需求，还最大限度地利用了LLMs已经学到的知识，从而增强了模型的

具有理解和执行与网络安全相关任务的能力，如自动威胁检测、漏洞识别和安全策略建议。

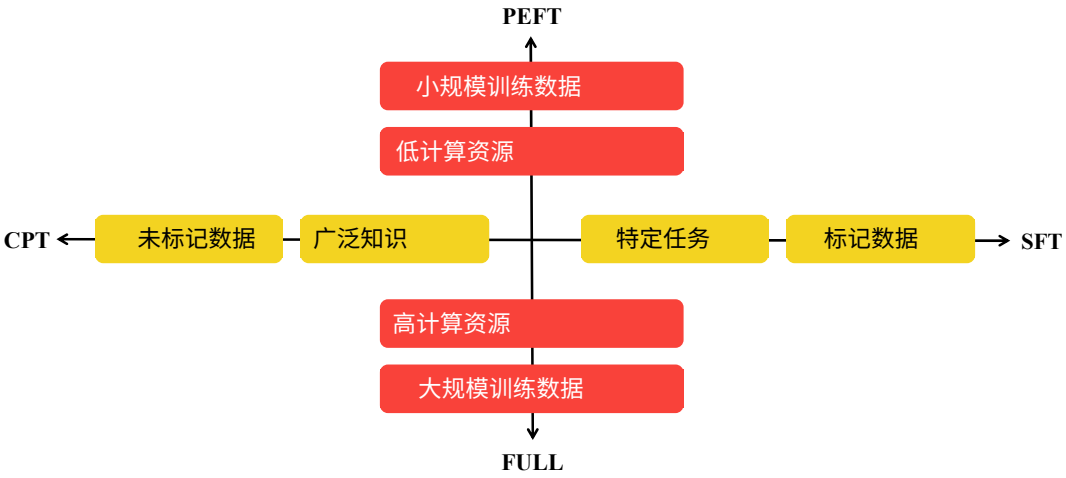


图5:领域LLM训练方法比较. *CPT*和 *SFT*提供了基于现有LLMs增强领域特定性能的方法，而 *FULL*参数训练和 *PEFT*代表了这些训练过程中不同的技术路径。

为了将通用LLMs适应网络安全，研究人员主要采用两种方法：持续预训练和监督微调。

持续预训练涉及使用大量未标记的领域特定数据对已经预训练的LLMs进行进一步训练[40, 41, 52, 53]。 这种方法旨在提高模型对领域知识的理解和应用，显着增强其在网络安全领域的广泛适用性。 持续预训练基于一个核心假设，即即使经过广泛的预训练，模型仍然具有进一步增强的潜力，特别是在特定领域或任务的性能方面。 该过程通常涉及几个关键步骤：首先，选择一个能够恰当代表目标领域特征的数据集；其次，确定持续预训练的策略；最后，执行预训练，根据需要调整模型架构或优化算法，以适应新的训练目标。

受监督的精细调整，另一方面，使用标记的领域特定数据进行训练，可以直接优化模型在特定网络安全任务上的性能[42,43]。 与持续的预训练相比，**SFT**更侧重于增强任务特定性能。 在**SFT**中，模型权重通过从特定任务损失函数计算的梯度进行细化。 该函数量化了模型预测与实际标签之间的偏差，从而促进了任务导向模式和细微差别的学习。 **SFT**依赖于高质量的人工注释数据，这是一组提示及其相应的响应的集合。受监督的精细调整对于像ChatGPT这样的LLMs尤为重要，这些模型被设计为遵循用户指令并在长篇文本中保持特定任务。 这种特定类型的精细调整也被称为指令精细调整。

在持续预训练和SFT的背景下，研究人员可以选择使用全参数精细调整或参数高效精细调整。

全参数微调是一种传统方法，在训练过程中调整整个模型的参数。
这使得模型能够完全适应并专门针对目标领域的细微差别。 通过优化所有参数，模型可以潜在地实现特定任务或数据集的最佳性能。 然而，这种详尽的参数更新需要大量的计算能力和时间，从效率和可扩展性的角度来看，尤其是随着大型语言模型的增加，这带来了挑战。

相反地， **PEFT**方法只微调少量（额外的）模型参数，同时冻结预训练的大型语言模型的大部分参数，从而大大降低了计算和存储成本。 这也有助于可移植性，用户可以使用PEFT方法调整模型，获得几MB的小检查点，而不是完全微调的大检查点。 PEFT方法备受青睐，因为它们使用户能够在只有少量可训练参数的情况下获得与完全微调相媲美的性能。 有几种PEFT方法，例如适配器微调，前缀微调，提示微调，LoRA，QLoRA等：

适配器调整 [54] 在变压器架构中的多头注意力和前馈层之后插入适配器，
在微调期间仅更新适配器中的参数，同时保持模型其余参数不变

冻结。*P*调整 [55] 通过引入可训练提示令牌自动学习最佳的任务特定提示嵌入，消除了手动提示设计的需要，并通过添加锚点令牌可能提高性能。前缀调整 [56] 保持语言模型参数冻结，并优化称为前缀的小型连续任务特定向量。提示调整 [57] 通过反向传播和合并标记示例学习软提示，为特定任务进行微调。 *LoRA* [58] 是一个可以插入变压器架构中的小型可训练子模块，涉及冻结预训练模型权重并将可训练的低秩分解矩阵注入变压器架构的每一层，大大减少了下游任务的可训练参数数量。 训练完成后，低秩分解矩阵的参数与原始LLM的参数结合使用。 *QLoRA* [59] 是LoRA的进一步优化，通过将梯度反向传播到一个带有冻结的4位量化预训练语言模型的低秩适配器，大大减少了微调的内存需求，同时几乎与完全微调相媲美。

通过整合这些技术，研究人员可以选择适当的方法来构建适合网络安全领域特定需求和条件的LLMs，如图5所示。 此外，新兴技术还为构建网络安全LLMs提供了见解。例如，模型编辑技术[60, 61]可以精确修改LLMs，以整合网络安全知识而不会对其他知识产生负面影响。 通过设计有效的提示来引导LLMs产生期望的输出，即提示工程[62, 63, 64]，可以缓解构建网络安全LLMs所需的训练数据和资源瓶颈。

3.2 通过评估网络安全能力选择基础模型构建领域LLM

如上所述，从头开始训练网络安全LLM是具有挑战性的。 一般做法是选择通用的LLM作为基础模型，然后进行微调。然而，如何在各种LLMs中选择适当的基础模型呢？ 基本思路是选择具有强大网络安全能力或在特定安全任务中表现良好的LLM。 这些模型更擅长理解和解决与安全相关的问题。 现有的LLM网络安全能力评估可以分为三个主要类别：网络安全知识、安全代码生成和IT运营能力。

网络安全知识评估侧重于评估模型对网络安全概念的理解以及提供关于安全威胁和缓解策略的准确信息。 **CyberBench** [65] 出现为一个领域特定的、多任务基准测试工具，用于评估LLMs在网络安全任务中的能力。作为网络安全领域LLMs的基准套件，**CyberBench**提供了一种通用但一致的方法，减轻了以前在该领域内评估LLMs时遇到的限制。 **SecEval** [66] 是为评估LLMs中的网络安全知识而创建的。它提供了超过2000个跨越9个领域的多项选择题：软件安全、应用安全、系统安全、网络安全、密码学、内存安全、网络安全和渗透测试。通过促进对十个最先进的基础模型在网络安全领域中表现的评估，这项研究为他们的表现提供了新的见解。 通过将专家知识与LLMs的合作相结合，**Norbert** T等人[45] 创造了**CyberMetric**基准数据集，其中包含10,000个旨在评估网络安全领域内各种LLMs的网络安全知识的问题。 此外，**Sec**QA [67]，一个由GPT-4基于"计算机系统安全：成功规划"教材生成的多项选择题数据集，旨在专门评估LLMs对安全原则的理解和应用。 **Sec**QA不仅提供了两个复杂度层次的问题，而且作为一个评估工具，还促进了LLMs在需要更高安全意识的环境中的应用进展。

安全代码生成测试模型生成的代码不仅功能正常，而且符合安全最佳实践，旨在最小化漏洞。M**ani**sh B等人[46]引入了一个名为Cyb**erSec**Eva**l**的安全编码基准，旨在评估LLMs生成代码时潜在的安全风险和促进网络攻击的倾向。 通过评估包括L**l**a**ma** 2、C**ode** L**l**a**ma**和O**pen**AI的GPT在内的七个模型，Cyb**erSec**Eva**l**有效地指出了关键的网络安全风险，并为模型改进提供了实用的见解。

Catherine T等人[47]引入了LLMSecEva*l*，这是一个包含150个基于MITRE的Top 25常见弱点枚举（CWE）排名中各种漏洞叙述的自然语言提示的数据集。 通过将LLMs生成的代码与每个提示的安全实现示例进行比较，LLMSecEva*l*可以评估LLMs生成的代码的安全性。Siddiq M等人[68]提出了SecurityEva*l*，专注于对生成代码模型的安全评估，以防止生成易受利用的代码，避免开发人员潜在的误用。 该数据集包括130个样本，涵盖75种漏洞类型，映射到CWE。Kame*l* A等人提出PythonSecurityEva*l* [69]，这是从Stack Overf*l*ow的真实场景中收集的真实数据集，用于评估LLMs生成安全Python代码的能力以及修复安全漏洞的能力。 DebugBench [70]，包含4,253个实例，涵盖了C++、Java和Python中的四个主要错误类别和18种次要类型。这一全面评估阐明了LLMs在自动调试中的优势和劣势，标志着对它们在实际编码场景中适用性和限制的重要进展。

IT运营能力评估模型在管理和保护IT基础设施方面的熟练程度，包括网络安全态势感知、安全威胁分析和事件响应。Yukai M等人[71]引入了NetEval，这是一个旨在衡量LLMs在多语境下NetOps中的常识和推理能力的评估集。 NetEval包括5,732个与NetOps相关的问题，涵盖了五个不同的NetOps子域。 通过NetEval，研究人员系统评估了26个公开可用LLMs的NetOps能力。此外，OpsEval [72]包含7184个多选题和1736个英文和中文问题回答格式，旨在用于故障根本原因分析、操作脚本生成和警报信息总结等任务，全面评估LLMs在IT运营任务中的表现。

评估LLMs的网络安全能力不仅指导在微调过程中选择基础模型，还表明通用LLMs具有一定的网络安全能力。 这支持直接利用LLMs（无需进行微调）来辅助网络安全应用的可行性，如第4节所讨论的。 此外，这些研究帮助研究人员和开发人员认识到LLMs在网络安全领域的局限性，为推动人工智能朝着更高标准和更专业化的安全发展提供方向。

3.3 为网络安全领域进行微调的大型语言模型

研究人员利用上述技术和基础模型定制了LLMs，以解决网络安全领域的特定问题。 这些努力突显了整合领域特定知识以增强语言模型能力的巨大潜力，尤其是对于包括漏洞检测、故障定位、程序修复等关键应用。

漏洞检测涉及识别和分类软件代码中潜在的安全弱点。Alexey S等人[73]专门为漏洞检测任务对WizardCoder [74]进行了Lora微调，重点关注Java函数是否包含漏洞的二元分类。Ferrag M等人[48]对FalconLLM [4]进行了部分参数微调，得到了SecureFalcon，可以以高达96%的检测准确率区分易受攻击和非易受攻击样本，并进一步提出了使用FalconLLM修复漏洞的方法。Aidan Y等人[75]引入了LLMAO，一种基于新语言模型的故障定位方法，在CodeGen [76, 77]上添加了双向适配器层，使模型能够学习代码的双向表示并预测代码行中缺陷的概率。

通过LLMs努力生成安全代码，旨在增加自动生成代码的安全性，减少漏洞风险。
Storhaug A等人[78]提出了一种创新方法，称为受漏洞约束的解码，该方法在模型训练过程中集成了漏洞标签。 在解码过程中避免生成带有标记漏洞的代码显着减少了易受攻击代码的生成。 在GPT-J上进行微调[79]显示出合成代码漏洞显着减少。Jingxuan H等人[80]专注于通过指令调整改进LLMs生成代码的安全性。 使用包含安全和不安全程序的数据集进行监督微调，将Code*e*LLama[34]转换为Safe*e*Code*e*r，从而在各种流行的LM和数据集中实现安全性的显着提升（约30%），同时保持实用性。

自动程序修复旨在在无需人工干预的情况下自动修复软件错误。André S等人[28]提出了一种名为RepairLLaMA的新程序修复方法，通过将Lora微调应用于CodeLlama，显着提高了LLMs的程序修复能力。它在Java基准测试Defects4J和HumanEval-Java上优于GPT-4。

二进制是计算机代码的最基本形式，重要的是要了解它的含义以及如何使用。Nan J等人 [81]将LLMs的好处带到了二进制领域，通过在专门的二进制代码预训练语料库和新任务上持续训练StarCoder [35, 36]，导致了Nova和Nova$^+$的开发。 在SFT之后，增强型LLMs有效地解决了特定任务，如二进制代码相似性检测、二进制代码翻译和二进制代码恢复。

IT运营负责维护例行任务和其他维护基础设施以支持其他服务的活动。Hongcheng G等人 [82]描述了为IT运营开发的专门LLM，命名为Owl，通过在收集的Owl-Instruct数据集上进行监督微调Llama实现。 Owl在IT相关任务中超越了现有模型，并在Owl-Bench基准测试中展现了有效的泛化能力。

网络安全知识助手有助于提高用户的安全意识，并通过与用户的互动帮助用户抵御网络攻击。杰・Z等人 [29]提出了Hackmentor，通过开发针对网络安全领域的指令和对话数据集，并基于该数据集对Llama和Vicuna [3]进行了LoRA微调，有效展示了LLM在网络安全应用中的广泛潜力。

这些研究显示了LLM在网络安全领域的重要潜力，不仅验证了通过SFT和持续预训练来调整LLM的有效性，还为未来与网络安全相关的研究开辟了新的途径。

Q1的答案：对于研究人员来说，通过使用诸如CPT和SFT等方法，通过使用网络安全数据调整通用LLM构建网络安全LLM是一条可行的技术路线，实施技术取决于特定的应用场景、资源可用性和预期的性能改进水平。

4 RQ2: LLM在网络安全中的潜在应用是什么?

组织成不同的类别，随后的章节审查了LLM在网络安全中的多方面贡献，从威胁情报和代码审查到漏洞发现和程序修复。 通过阐明每个主题集群中的关键进展，本综述旨在提供一个全面的视角，强调LLM整合所支持的信息安全不断发展的格局。

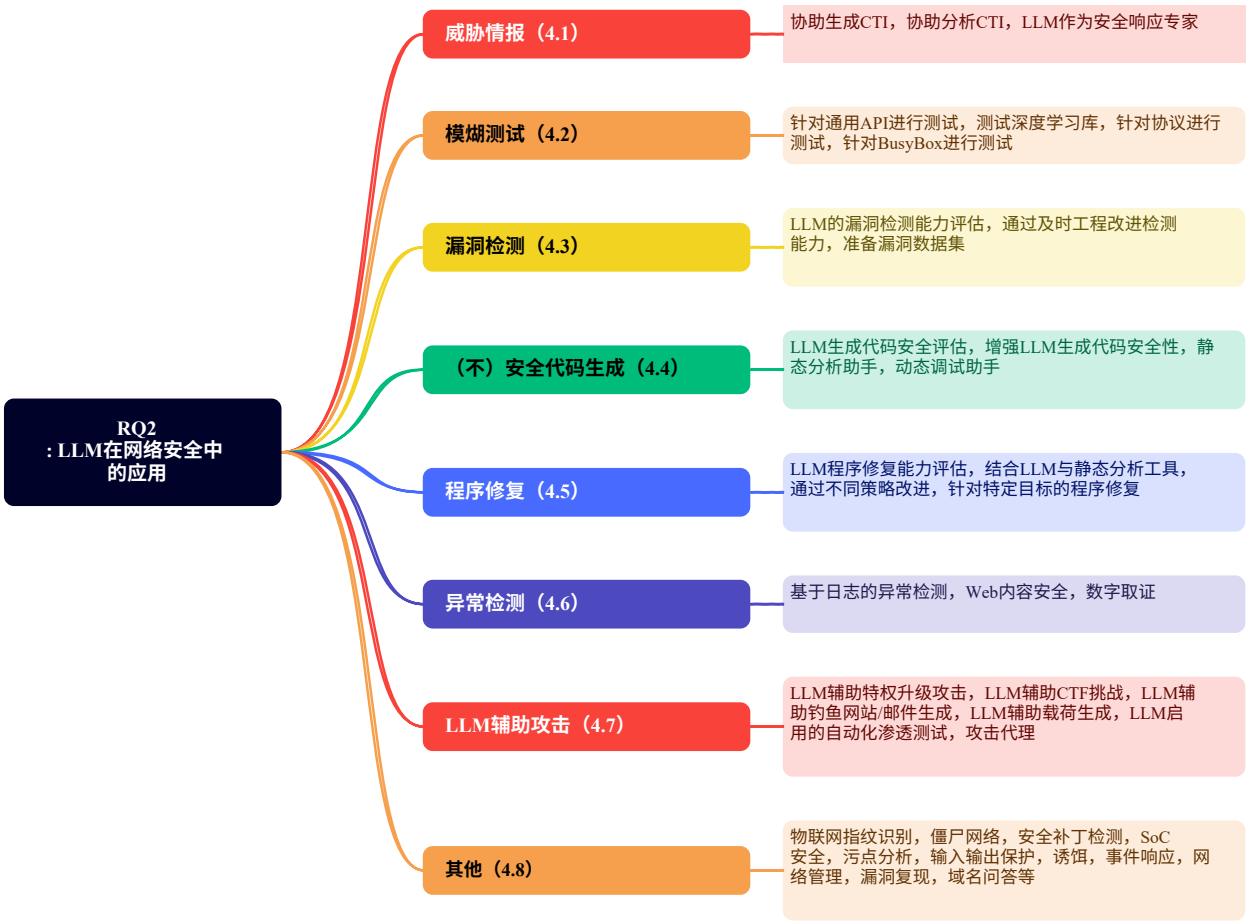


图 6: **RQ2** 的概述。

4.1 威胁情报

由于大型语言模型在自然语言处理（NLP）任务中展示出的出色分析和总结能力，一些研究人员正在努力利用它们来协助生成和分析网络威胁情报（CTI）。

Shaswata M等人[83]提出了一个名为LocalIntel的框架，旨在通过允许LLM在查询全球和本地知识数据库后总结知识，为用户提供可靠的威胁情报。全球知识主要指来自CWE和CVE的关于网络安全威胁的详细报告，而本地知识由组织定制用于实际目的，从而补充全球知识。Filippo P等人[84]也进行了类似的工作，利用LLM从广泛的知识库中提取安全知识并自动生成报告。 一些类似的努力如下。Reza F等人[85]利用LLM生成网络攻击描述，通过使用从ATT&CK和CAPEC收集的信息对模型进行微调。然后，他们比较了经过微调的LLM（例如BERT）与直接使用的LLM（GPT-3.5）在描述攻击方面的性能。 在另一项工作中，Reza F等人[86]研究了LLM在网络安全中解释和总结网络攻击战术、技术和程序（TTPs）的应用。 它比较了仅编码器（例如RoBERTa）和仅解码器（例如GPT-3.5）模型在TTP分析中的有效性，并引入了检索增强生成（RAG）以增强解码器模型而无需微调。 研究发现，RAG通过提供相关背景显着改善了对TTP的解释，突显了LLM在威胁情报中的潜力。Tanmay S等人[87]讨论了LLM自动化分析和总结软件供应链安全漏洞的能力。 他们评估了LLM在复制69个故障的手动评估中的表现，重点关注分类准确性。 结果表明，LLM显示出潜力，特别是在有全面数据的情况下，但在这个微妙领域仍无法取代人类分析师。Samaneh S等人[88]评估了各种大型语言模型在威胁情报领域的性能，包括ChatGPT、GPT4all、Dolly等。 利用基于Twitter的开源情报（OSINT）数据集，研究评估了这些聊天机器人在二元分类和命名实体识别（NER）任务中的能力。 虽然LLM在二元分类中表现出有希望的结果，但它们在网络安全实体识别的NER方面的有效性有限，突显了LLM技术在增强CTI应用方面的进一步发展的需求。 特别是对于数字取证，Gaetan M等人[89]提出了一种自动生成报告的方法。 他们检查了取证报告的结构，以识别常见部分，并评估了LLM在生成这些部分方面的可行性。 通过案例研究方法，该文章评估了LLM在创建取证报告的不同部分方面的优势和局限性。

特别是考虑到大多数威胁情报提供商提供的信息是以非结构化格式，Giuseppe S等人[90]和Yuelin H等人[91]提出了创新性的解决方案，以解决从非结构化信息中提取有用信息的普遍问题。 前者设计了一个名为aCTIon的框架，其中包括下载/解析原始报告，利用LLM提取有用信息，并按照STIX [92]标准导出结构化报告。 后者构建了用于非结构化威胁情报的知识图，并对LLM进行了微调，以自动化这一任务。

除了从大量文本中提取有价值的信息外，报告去重也是威胁情报领域的一个重要研究点。Ting Z等人[93]依靠LLM来缓解漏洞报告去重的问题。 他们利用LLM作为一个中间步骤，通过充当识别关键词的角色，提高了REP [94]的性能（REP是一种传统方法，用于衡量漏洞报告之间的相似性）。

此外，已经有一些研究尝试将LLMs用作经验丰富的安全响应专家。YuZheng L等人[95]使用LLM作为关于通过及时工程减轻漏洞的建议提供者。 他们设计了一个系统，可以在用户输入漏洞描述后用于检索相关的CVE和CWE信息。 LLM的减轻建议是这个系统的一个子部分。Mehrdad K等人[96]将LLM不仅视为具有专业知识的问答助手，还根据用户的描述执行操作（例如，指示主机计算机的入侵检测系统阻止特定IP）。为了增强网络安全中的战略推理，[97]引入了Crimson，这是一个利用大型语言模型将CVE与MITRE ATT&CK技术相关联以提高威胁预测和防御的系统。 他们的关键思想涉及到一种称为检索感知训练（RAT）过程，该过程使LLMs精细化生成精确的网络安全策略，显着减少错误和幻觉。 通过整合实时数据检索和领域特定的微调，Crimson提升了模型的可解释性和战略连贯性，为网络安全威胁情报提供了一种积极的方法。

4.2 模糊

传统的模糊技术虽然在发现软件漏洞方面有效，但存在固有的局限性，可能会影响其效率和有效性。 一个重要的缺点是传统的模糊器在很大程度上以随机或半随机的方式运行，这可能导致一种耗时且有时无效的测试方法，因为它们可能无法探索所有可能的执行路径。 此外，要变异的种子通常是由人类精心制作的，这是耗时的。 尽管这些问题多年来已经得到研究，并且有许多方法可以缓解它们，但LLM的出现为模糊领域提供了一种全新的思考方式

LLM模糊相对于传统方法的优势是什么？

Ying Z等人[98]评估了Ch*at*GPT在生成测试用例时的性能（无需调整），并将其与两种传统测试工具（SIEGE和TRANSFER）进行了比较。他们的实验表明，当给出漏洞的详细描述、可能的利用方式和代码上下文时，LLM优于传统方法。

以下是LLM相对传统工具的优势描述。 最重要的因素之一在于LLM的出现导致了从随机突变到有目的突变的引导。Jie H等人[99]在传统的灰盒模糊测试中添加了基于GPT的种子突变器，从种子池中选择种子，并从ChatGPT请求变体以生成更高质量的输入。 另一个因素是LLM具有良好的跨编程语言理解能力，因此能够在多种编程语言中执行测试任务。Chunqiu S X等人[100]充分利用了LLM对不同编程语言的理解能力。大多数传统方法只能对特定编程语言进行模糊测试，而基于LLM的模糊测试可以涵盖不同的语言。 他们使用名为Fuzz-Loop的方法测试了6种语言代码（C、C++、Go、SMT2、Java和Python），该方法可以自动变异测试用例。 大多数传统模糊测试方法无法在所有代码中实现高代码覆盖率，而掌握代码逻辑的LLM可以针对那些低覆盖率代码生成更有针对性的测试用例。 例如，Caroline L等人[101]使用Codex针对低覆盖函数生成测试用例，当SBST（基于搜索的软件测试，一种传统的模糊测试方法）达到覆盖率平台时。 具体来说，Codex生成的原始字符序列被反序列化为SBST的内部测试用例表示，以利用SBST的突变操作和适应性函数。

针对不同的测试对象使用特定的模糊测试策略。

根据被测试的内容，当使用LLM时，策略需要进行一定程度的调整。对于测试通用**API**Cen Z等人[102]研究了LLM在生成调用代码方面的有效性。 该研究将基于LLM的生成与传统程序分析方法进行比较，并发现LLM可以自动产生大量有效的模糊驱动程序，减少了手动干预。 该研究引入了查询策略、迭代改进以及示例的使用来提升LLM的性能。 尽管一切都是关于测试**API**，但是对于测试深度学习库的策略需要进行修改。 调用深度学习库的程序通常对张量维度有严格要求，否则模糊测试将执行大量无意义的测试。

Yinlin D等人 [103]提出了Ti*ta*nFuzz，一个为深度学习库生成测试用例的工具。他们的训练语料库包含大量调用DL库API的代码片段，从而可以隐式学习语言语法/语义和复杂的DL API约束，以有效地生成DL程序。另一项工作，FuzzGPT
[104]，由Yinlin D等人进行的研究也是关于模糊DL库的研究。 与先前的工作相比，FuzzGPT专注于使用历史错误触发的代码片段来指导LLM生成的测试用例。

除了上述研究，我们还收集了一些针对其他测试对象的论文。针对协议的测试。Ruijie M等人 [105]讨论了如何在没有可读的机器协议规范的情况下找到协议实现中的安全漏洞。 他们用大量人类可读的协议文档训练了LLM，并要求LLM对协议模糊（例如H**TTP**）的交互消息进行变异。针对**B**usy**B**ox的测试。 专门针对**B**usy**B**ox，这是Linu**x**设备中普遍使用的实用程序，Asmita等人 [106]引入了两种方法：利用LLMs生成针对特定目标的初始种子进行模糊测试，这显着提高了识别崩溃和潜在漏洞的效率；以及崩溃重用，利用先前获取的崩溃数据来简化对新目标的测试过程。

4.3 漏洞检测

本节提供了关于利用LLMs进行漏洞检测的关键研究论文概述。通过审查这些作品，我们旨在阐明利用LLMs增强网络安全的进展、挑战和未来方向。

*(*在本节中，我们模糊了"漏洞"和"软件缺陷"的概念*)*

LLMs是否有能力检测漏洞?

以下几篇论文对这个问题进行了基础研究。 尽管它们在这个问题上的结果可能因为一些未知原因（例如，可能使用不同的数据集）而有所不同，但总体上它们都表明LLM对漏洞检测这个问题具有很好的前景。

在早期阶段，Anton C等人[107]进行了测试，以评估GPT-3和GPT-3.5是否能够识别Java代码中的一些已知CWE漏洞。 结果显示在漏洞检测任务中的应用并不那么好，需要进一步的改进和研究。 在另一项工作中，Moumita D等人[108]使用LLMs（包括GPT-3.5、CodeGen和GPT-4）分析了几种常见的漏洞（例如SQL注入、溢出）。
结论是，尽管它确实确认了LLM具有检测漏洞的能力，但误报率很高。 然而，Marwan O [109] 在各种易受攻击代码基准数据集上对GPT进行了微调

以便检测软件漏洞。 结果表现出良好的性能。 同样，Avishree K等人 [110] 得出结论，LLM（包括GPT-4和 CodeLlama）通常能够比现有的静态分析和基于深度学习的工具更好地执行漏洞检测。 通过精心设计的提 示，可以在合成数据集上获得理想的结果，但在更具挑战性的真实世界数据集上性能会下降。Rasmus I T J 等人 [111] 比较了广泛范围的开源模型和专有模型在辅助漏洞发现时与Python代码片段的性能。 他们的研 究表明，LLM可以有效地用于增强代码审查的效率和质量，特别是在检测软件代码中的安全问题方面。Alex ey S等人 [73] 对WizardCoder进行了漏洞检测任务的微调，并调查了遇到的性能限制是否是由于类似Code BERT模型的有限容量。 他们的研究结果表明，LLM在漏洞检测方面有着光明的未来。Haonan L等人 [112] 提出了LLift，一个利用LLM辅助静态程序分析的框架，特别是用于检测初始化前使用（UBI）缺陷。 LLift 与静态分析工具和LLM进行接口，展示了在真实场景中50%的精度率，并识别了Linux内核中先前未知的13 个UBI错误。

通过不同的策略提高检测能力。

许多研究人员在向LLMs提供代码并直接要求答案之前，会采取不同的策略。 部分研究人员认为仅仅提供代 码是不够的，也就是说，代码需要进一步预处理或者需要为LLM提供更多信息以进行漏洞推理。 Jin W等人[113]并没有直接向模型提供代码，而是进行了代码序列嵌入（CSE），将代码的AST、DFG和CFG组合作为 模型的输入。 他们借助Conformer机制（Transformer的改进架构[114]）捕获了输入的语义信息。Chenyu an Z等人[115]不仅向GPT提供了代码，还提供了API调用序列和数据流图。Atieh B等人[116]进行了类似的 实验。 他们通过给模型不同级别的信息进行比较（直接要求漏洞点、在要求LLM定位之前提供一些CWE信 息，以及在要求LLM定位之前告诉LLM代码中存在哪些漏洞）。Noble S M等人[117]专注于Android平台的 漏洞，并比较了LLM在三种条件下的性能：直接要求LLM找到漏洞、在询问之前提供漏洞摘要，以及在仅提 供APK核心（AndroidManifest.xml和MainActivity.java）后授予LLM请求任何所需文件的权限。

除了上述努力外，研究人员提出了许多创新的想法来提高LLM的漏洞检测能力。Sihao H等人[118]提出了一 个名为GPTLENS的创新的两阶段框架，其中包括两个对抗性代理角色：审计员和评论员。 审计员的角色在 生成阶段执行，其主要目标是识别智能合约中的潜在漏洞。 评论员的角色在识别阶段执行，其主要目标是 评估审计员生成的漏洞。Zhihong L等人[119]使用传统算法（TF-IDF和BM25）将待分析的代码与漏洞语料 库中的代码进行匹配以进行相似性分析。 待分析的代码以及语料库中的相似代码将一起提供给LLM。 基于 上下文学习的理念，LLM可以更好地分析其是否属于这种漏洞。 专门针对智能合约中的漏洞检测，Yuqiang S等人[120]提出了一个名为GPTScan的工具。GPTScan首先解析智能合约项目以确定函数的可达性，只有 潜在易受攻击的函数将被保留。 然后，GPTScan使用GPT将候选函数与预定义的漏洞类型进行匹配。 最后 ，GPTScan要求GPT验证漏洞。 为了提高LLM对漏洞的推理能力，Yuqiang S等人[121]提出了LLM4Vuln， 该方法将LLM的漏洞推理能力与其他方面（例如主动寻求额外信息，应用相关漏洞知识和按照指示输出结构 化结果）分开。 他们允许LLM请求获取有关目标代码的额外上下文信息。 此外，他们得出结论，输入到LL M的信息越多，并不意味着性能就会更好。例如，完整的漏洞报告，大量的调用上下文，提供过多信息可能 会导致分心。Zhenyu M等人[122]提出了一种名为MuCoLD的新方法，该方法模拟了软件中漏洞检测的多角 色代码审查过程。 通过扮演不同的角色，如开发人员和测试人员，LLM参与讨论以就漏洞的存在和分类达 成共识。 该方法以二进制判断和推理为初始点，通过迭代对话来完善评估。

目前的想法是检测特定程序中的漏洞，但也有研究利用LLM从漏洞中推断受影响的库列表。田宇C等人得出 结论，NVD中的许多漏洞报告没有列出受影响的库，或者列出了不完整或不正确的库名称，增加了第三方库 漏洞的风险。 他们提出了一种名为VulLibGen的方法[123]，旨在检测第三方库中的漏洞。 VulLibGen仅将 漏洞描述作为输入，并利用LLM的先验知识为给定漏洞生成受影响库名称列表。

与先前研究不同，裴宇L等人[124]提出了一种关于将ChatGPT应用于漏洞管理的方法，评估其在预测安全漏洞、评估严重性、修复漏洞和验证补丁正确性方面的能力，使用了大量数据集。 研究表明，虽然ChatGPT可以帮助识别和减轻软件安全威胁，但在漏洞优先级排序和补丁验证等任务方面仍需要改进。

数据集准备除了重新训练或微调模型所使用的方法外，数据集的构建也是一个重要的方面。

Yizheng C等人[125]提出了一个名为DiverseVul的新的易受攻击源代码数据集，其中包含18,945个易受攻击函数（涵盖150个CWE）和330,492个正常函数。 所有这些样本都是C/C++代码。
此外，他们讨论了11种不同的深度学习架构，并得出结论，尽管LLMs取得了成功，但模型仍然面临高误报率、低F1分数和难以检测复杂CWE的挑战。Norbert T等人[126]制作了一个包含112,000行C代码的数据集，详细标记了有关漏洞的信息（CWE编号、位置和函数名称）。 此数据集中的所有代码均由GPT-3.5生成。
Zeyu G等人[127]提出了一个名为VulBench的全面漏洞基准数据集，其中包括来自CTF挑战和真实应用程序的高质量数据，详细注释了每个易受攻击函数的漏洞类型和原因。

4.4 (不)安全的代码生成

之前的许多工作证实了大型预测模型确实具有良好的代码理解能力。
然而，生成的代码的安全性非常重要。 已经进行了许多研究，探讨LLMs是否能提供安全的代码。

评估LLM生成代码的安全性。

LLM生成的代码是否存在安全风险？Gustavo S等人[128]进行了一项实验，探讨了在LLM的帮助下由本科计算机科学专业学生编写的代码是否存在安全风险。 参与者的任务是在C语言中实现一个单链表的'购物清单'结构，并分为两组：'对照组'（无Codex LLM访问权限）和'辅助组'（有Codex LLM访问权限）。 结果显示，当用作代码助手时，LLM并不会显着增加引入安全漏洞的风险。Florian T等人。

[129]进行了一项实证研究，调查了由大型语言模型（LLMs）生成的代码中的错误，重点关注三个主要模型：CodeGen、PanGu-Coder和Codex。研究在333个收集的错误中识别出10种不同的错误模式，并通过对34名LLM从业者和研究人员进行在线调查验证了这些模式。

到目前为止，已经进行了许多研究来探索最先进的LLM生成的代码的安全性。Hammond P等人 [130]调查了GitHub Copilot生成的代码的安全性。研究人员设计了89种不同的场景供Copilot完成，结果产生了1,689个程序。 他们分析了这些程序的漏洞，特别关注了MITRE统计的前25个CWE。[131]深入探讨了LLMs在面向安全的程序分析中的潜力。 他们关注了两个代表性的LLMs，ChatGPT和CodeBERT，并评估它们在解决不同难度级别的分析任务（包括漏洞分析和错误修复、Fuzzing、汇编代码分析等）中的表现。Zhijie L [132]评估了ChatGPT生成的代码，重点关注正确性、可理解性和安全性。 通过使用LeetCode问题和CWE场景进行的实证研究，他们分析了ChatGPT生成的代码片段的质量以及其进行多轮对话以改进代码的能力。 结果显示，虽然ChatGPT可以生成功能正确的代码，但在复杂推理和保持代码安全方面存在困难。

另一方面，Mohammed L等人[133]提出了一个名为SALLM的框架，这是一个专门用于评估大型语言模型生成代码安全性的基准。SALLM由三个组件组成，一个描述Python程序文本的提示数据集，一个需要LLM不同解决方案的代码生成环境，以及利用Docker执行生成代码的系统化模型评估方法。 他们在5个LLM上测试了他们的方法。Jiawei L等人[134]专注于代码生成的质量评估。

现有基准通常只包含有限数量的测试用例。 为了解决这个问题，他们提出了EvalPlus，一个代码合成评估框架，通过自动化测试输入生成器（结合LLM和基于突变的策略）大幅扩展了评估数据集中的测试用例数量。Saad U等人[135]构建了一系列228个代码场景，并在自动化框架中分析了八个最先进的LLM，以确定LLM是否能可靠地识别与安全相关的漏洞。 他们指出当前的LLM在自动化漏洞检测任务中尚未表现令人满意，并列出了当前LLM表现出的一系列缺点。Alessio B [136]评估了ChatGPT-3.5在生成代码（包括代码安全性）方面的性能，并分析了这种能力在10种编程语言上的潜力。

LLM是否知道生成的代码是否安全?

Raphaël K等人[137]进行了一系列实验，验证了LLM生成的代码的安全性，并在不同场景中发现了生成的代码的漏洞。 尽管LLM可能在被要求审查后识别出代码生成本身的漏洞，但实验表明，除非明确指示这样做，否则LLM仍会生成不安全的代码。他们还在研究中提到了一个关键问题，即由于深度神经网络的不可解释性，对LLM的重复提问往往会导致不同的答案（代码是否不安全），而无法找到最大化成功识别的策略。

更直接地，Jingxuan H等人[138]尝试通过一些机制指定LLM生成安全或不安全的代码。
他们提出了一种名为svGen的方法，根据用户的需求使LLM生成安全或不安全的代码。 除了对生成的代码的描述，他们还引入了特定属性的连续向量，称为前缀，这些向量序列与LLM的隐藏状态的形状匹配。 这些前缀经过优化，通过提供初始隐藏状态来影响LLM的生成过程，使代码朝着满足所需安全标准的方向发展，而不修改LM的基础权重。"

为了增强LLM生成代码的安全性，Jingxuan H等人[80]引入了SafeCoder，这是一种创新的指令调整方法，通过使用从GitHub自动流水线收集的高质量数据集，将标准指令调整与安全特定的微调有效结合起来。SafeCoder显着提高了代码安全性，而不会影响LLM在各种任务中的效用，展示了其在提升LLM生成代码安全性方面的多功能性和适用性。

LLMs作为静态分析助手。

Hammond P等人[139]探讨了LLM的应用，例如OpenAI的Codex，在逆向工程领域，特别是在理解软件功能和从代码中提取信息方面。 LLMs主要用于分析由逆向工程工具（如Ghidra）提供的类似C的代码的功能。这些C代码是通过反编译过程从二进制文件中获得的。 在逆向工程中，反编译也是一项重要任务。Hanzhuo T等人[140]引入了专为反编译而定制的LLM，专注于将编译后的机器代码转换回人类可读的源代码。 他们在大量C代码和汇编代码对上对一个名为DeepSeek-Coder的LLM进行了微调，并通过重新编译和执行反编译代码来评估他们工作的性能。

崇州F等人 [141] 探讨了大型语言模型在代码分析任务中的潜力和局限，特别是在处理混淆代码的情况下。 在他们的实验中，他们还进行了允许LLMs生成去混淆版本的代码的实验，也就是说，从混淆代码中恢复原始、更易读的代码。

简宇Z等人[142] 关注如何通过模糊测试提高LLM对程序的语义理解。
他们的核心思想是，程序及其基本单元（即函数和子程序）被设计为展示不同行为并在不同输入下提供可能的输出。 因此，通过模糊测试，各种输入会触发代码的不同功能，从而帮助LLMs理解程序。大卫N等人 [143] 提出了ASTxplainer，这是一种用于编码中的大型语言模型的可解释性方法。 它将标记预测与抽象语法树（AST）节点对齐，从而实现对模型预测的详细评估和可视化。 ASTxplainer包括AsC-Eval用于结构性能估计，AsC-Causal用于因果分析，以及AsC-Viz用于可视化，能够更好地解释LLM。

裴Y等人 [144] 着重于如何利用大规模语言模型来辅助恶意软件的动态分析。
研究的核心思想是使用GPT-4为每个API调用生成解释性文本，然后使用预训练的语言模型BERT根据先前分析生成一系列API序列以执行。 这种方法理论上可以在生成过程中为所有API调用生成表示，而无需训练数据集。Himari F等人 [145] 利用大型语言模型，特别是ChatGPT，分析勒索软件通信的语言和战略要素。 通过检查一系列勒索软件样本，研究确定了在勒索说明中使用的模式和策略，揭示了勒索软件策略的演变，其特点是使用复杂的语言和心理操纵。P.V. Sai C等人 [146] 还讨论了LLMs在制定针对勒索软件的政策方面的潜力和挑战。Nusrat Z等人 [147] 利用GPT-3和GPT-4分析JavaScript包，在npm生态系统中检测潜在的恶意软件。 该研究介绍了SocketAI Scanner，这是一个利用迭代自我完善和零射击角色扮演思维链提示技术的多阶段工作流程，以增强模型识别代码中恶意意图的能力。 通过将LLMs的性能与静态分析工具进行比较，论文证明LLMs可以有效地准确识别恶意软件，并且误报率较低。

LLMs作为动态调试助手。

Runchu T等人 [70]引入了DebugBench，这是一个用于评估大型语言模型在编程中调试能力的基准。 它包括C++、Java和Python中各种bug类别的4253个实例。该基准是通过从LeetCode收集代码片段，使用GPT-4植入bug，并进行严格的质量检查构建的。Zhe L等人 [148]解决了移动应用程序自动化图形用户界面（GUI）测试的挑战。 他们提出了一种称为GPTDroid的新方法，该方法将GUI测试问题制定为问答（Q&A）任务，LLM被要求通过传递GUI页面信息与移动应用程序交流以引发测试脚本。 这些脚本被执行，应用程序反馈被迭代地传递回LLM以指导进一步的探索。Baleegh A等人 [149]提出了一种名为FLAG的方法，用于帮助人类调试人员识别和定位代码中的安全和功能性bug。 FLAG通过输入一个代码文件，并重新生成该文件中的每一行进行自我比较来工作。 它将原始代码与LLM生成的替代方案进行比较，以标记显着差异作为异常进行进一步检查。

4.5 程序修复

软件开发生命周期深受错误的影响，它们的检测和解决需要巨大的开支。 研究人员积极寻找新的方法来自动识别和纠正错误/漏洞。

对现有LLMs在程序修复方面的评估。

与最先进的不同LLMs（开源或专有）相比，许多研究已经评估了它们在程序修复方面的能力。Julian Ar等人 [150]探讨了OpenAI的Codex模型在自动程序修复（APR）领域的应用，特别是其在软件中定位和修复错误的能力。 他们使用了QuixBugs基准（包含Python和Java中的40个错误）来评估Codex在APR任务上的表现。 尽管没有重新训练，Codex的性能超过了许多现有的APR技术。Dominik S等人 [151]进行了类似的工作。他们都在QuixBugs基准上测试了LLM进行自动程序修复。 在这项工作中，评估了ChatGPT而不是Codex。Jan N等人 [152]讨论了大型语言模型Gemini在自动修复软件漏洞方面的应用，特别是对C/C++、Java和Go代码中由sanitizer工具发现的漏洞。 作者认为，虽然成功率可能看起来适度，但随着时间的推移，它有潜力节省大量的工程工作。Jiaxin Y等人 [153]评估了三个LLM：Gemini Pro、GPT-4和GPT-3.5，在具有实际代码审查中确定的安全缺陷的代码上。 研究结果表明，GPT-4的表现相对较好，但所有LLMs在响应简洁性、清晰度和准确性方面都有显着的改进空间。Chunqiu S等人 [154]选择了九个LLM，并将它们与传统的自动程序修复方法进行比较，展示了大型语言模型在这一领域的卓越表现。

Hammond P 等人 [155]研究了大型语言模型在代码零日漏洞修复中的潜力。
通过对合成、手工制作和真实世界安全场景进行大规模实验，他们表明，虽然LLMs在修复简单场景方面表现出潜力，但在处理更复杂的真实世界示例时却遇到困难。 本文有助于了解LLMs在网络安全中的能力，并鼓励进一步探索它们在漏洞修复中的应用。Yi W等人 [156]比较了LLMs和基于深度学习的APR模型在修复Java漏洞方面的能力。 他们评估了五个LLMs（Codex、CodeGen、CodeT5、PLBART和InCoder）、四个经过精细调整的LLMs以及四种基于深度学习的APR技术在两个真实世界Java漏洞基准（Vul4J和VJBench）上的性能。他们还设计了代码转换来解决Codex面临的训练和测试数据重叠问题，并创建了一个新的Java漏洞修复基准VJBench，以更好地评估LLMs和APR技术。

将大型语言模型与静态分析工具结合起来。

一些研究并未单独使用LLMs进行程序修复，而是将它们与传统程序分析工具结合起来，以提高这些工具的效率。Kamel A等人[69]提出了一种称为反馈驱动安全补丁（FDSP）的新方法，该方法将来自静态代码分析工具Bandit的反馈传递给LLM。LLM将生成潜在解决方案来解决安全漏洞。 每个解决方案以及易受攻击的代码都会被发送回LLM进行验证。Matthew J等人[157]提出了一个名为InferFix的程序修复框架，该框架整合了最新的静态分析器，用于修复关键的安全和性能漏洞。 InferFix由检索器和生成器组成。 检索器旨在搜索语义上等效的错误和相应的修复，而生成器则在通过添加错误类型注释和语义上相似的修复来增强提示的错误修复数据上进行微调。

通过不同的策略提高修复能力。

为了提高LLM在程序修复任务上的性能，研究人员进行了一些改进。
David D 等人 [158] 在包含C代码漏洞的数据集上对LLM进行了微调。 他们专门设计了一个

提供给LLM的代码的结构化表示，包括需要修复的代码行号，漏洞描述（CWE描述），完整的源代码等。LLM的输出也是结构化的，可以直接修补，从而使代码可以在没有人为干预的情况下自动修复。

Xinyun C等人 [159] 提出了一种称为SELF-DEBUGGING的方法。模型能够通过观察执行结果并解释自然语言生成的代码来识别错误，而无需任何关于代码正确性或错误消息的人类反馈。Toufique A等人 [160] 探讨了自一致性的应用。
[161]（提高模型推理能力的方法）在程序修复中。 通过将提交日志（在Github上收集的错误修复提交）作为少样本提示中的推理路径，Self-Consistency允许LLM生成多样化的解决方案。 从多个样本中选择最频繁的解决方案以提高补丁准确性。Yuxiang W等人[162]提出了一个名为Repilot的程序修复框架。它从掩盖错误代码段开始，然后利用LLM生成候选补丁。 在生成过程中，Repilot咨询完成引擎以修剪不可行的标记，并在必要时主动完成代码。 这种方法提高了补丁的编译率和正确性，同时减少了生成过程中无效尝试的次数。Nafis T等人[163]提出了SecRepair，这是一个利用LLM在软件中检测和修复代码漏洞的系统。 它利用具有语义奖励机制的强化学习来提高模型生成准确代码注释和描述的能力，引导开发人员有效解决安全问题。Jiaolong K等人介绍了ContrastRepair，通过为LLM提供对比测试用例对（包括失败测试和通过测试），增强了基于对话的修复框架，以提供更精确的反馈。 关键见解是最小化生成的通过测试和失败测试之间的差异，有效地隔离错误原因。 ContrastRepair与ChatGPT交互，迭代生成补丁，直到产生合理的修复。与先前的函数级方法不同，Yuxiao C等人[164]研究了LLM在程序修复的存储库级方法上的性能，这需要考虑代码之间可能跨越多个函数或文件的交互和依赖关系。 在他们的工作中，他们提出了一个基准，RepoBugs，包括来自开源存储库的124个错误，以评估LLM在这种情况下的性能。

针对特定目标的程序修复。

我们还有一系列文章，这些文章是针对一些特定目标的程序修复。 M.Caner T和Berk S [165]提出了一个名为ZeroLeak的框架，探讨了如何利用大型语言模型自动生成修复代码来解决软件中的侧信道漏洞。 ZeroLeak指导LLM生成特定漏洞的补丁，零-shot学习。 生成后，这些补丁将通过动态分析工具进行检查，以确保它们不仅在功能上正确，而且在防止信息泄漏方面也是安全的。Sudipta P等。

[166]提出了一个名为DIVAS的新颖框架。该框架将用户定义的SoC规范映射到常见弱点枚举（CWEs），为验证生成SystemVerilog断言（SVAs），并强制执行安全策略。 DIVAS自动化了漏洞检测和策略执行的过程，减少了手动工作量，增强了SoC的安全性。Baleegh A等人。[167]构建了一个包含领域代表性硬件安全漏洞的语料库，并利用LLM自动修复其中包含这些漏洞的Verilog代码。Tan K L等人。[168]专注于大型语言模型的应用，特别是ChatGPT和Bard，在修复JavaScript程序中的安全漏洞方面。 利用2023年CWE前25名清单作为参考，选择与JavaScript相关的漏洞，以评估模型生成正确补丁的准确性。 他们的研究突显了LLM在JavaScript安全性方面的潜力，强调了它们在主要用于Web开发的语言中的表现。

4.6 异常检测

在这一部分中，我们主要指的是与机器学习中的异常检测有所不同，主要是指安全异常，如流量中的恶意流量，系统中的病毒文件，日志中的异常等。

基于日志的异常检测。

Egil K等人[169]测试了60个针对日志分析进行微调的语言模型，包括BERT、RoBERTa、DistilRoBERTa、GPT-2和GPT-Neo等不同架构的模型。结果显示，通过微调，这些模型可以有效地用于日志分析，特别是针对特定日志类型的领域适应性。 针对华为云上的服务日志，Jinyang L等人[170]提出了一个名为ScaleAD的框架，旨在为云系统中的日志异常检测提供准确、轻量级和自适应的解决方案。 当ScaleAD的Tri-based Detection Agent (TDA)检测到可疑的异常日志时，它可以向包含的LLM发出查询请求，以获取这些日志的验证。LLM通过理解日志内容的语义来确定是否是异常，并提供相应的置信度分数。Jiaxing Q等人[171]提出了一个名为LogGPT的日志异常检测框架。LogGPT框架由三个主要组件组成：日志预处理、提示构建和响应解析器。 日志预处理组件涉及将原始日志消息进行过滤、解析和分组，以便进一步分析。 响应解析器负责提取ChatGPT返回的输出，以便进一步分析和评估检测到的异常。Xiao H等人[172]进行了一项

类似的工作。 不同之处在于，他们通过引入一个顶级K奖励度量来对GPT-2进行微调，从而指导模型集中关注日志序列中最相关的部分，提高异常检测的准确性。Yilun L等人。
[173]提出了一种称为LogPrompt的在线日志分析方法。 他们设计了一个指定格式，使用LLM解析非结构化日志，并生成具有特定结构的报告。 然后利用思维链，上下文学习方法，LogPrompt逐步推理日志内容，为正常/异常生成理由。魏Z等人。[174]引入了LEMUR，这是一个先进的日志解析框架，通过集成熵采样实现高效的日志聚类，并利用LLMs进行语义理解，增强了日志分析。 LEMUR通过丢弃手动规则并专注于语义信息来解决传统解析器的局限性。 利用LLMs的语义理解，该框架准确区分参数和不变标记，从而在日志模板合并和分类方面实现了令人印象深刻的效率和最先进的性能。

网络内容安全。

基于网站内容，Tamás V等人[175]进行了一项研究，以检测恶意网址。为了适应网络内容过滤，知识蒸馏被用来将老师的知识传递给更小的学生模型。 他们通过老师模型对未标记的URL进行分类，生成标签。 学生模型通过老师生成的标签进行训练，取得了更高的准确性，参数显着减少，使其适用于恶意URL的检测。Michae*l* G等人[176]探索了LLMs在检测DDoS攻击中的潜力，检查了LLMs在两个数据集（CICIDS 2017和Urban IoT数据集）上的性能。在CICIDS 2017中，通过少样本学习，他们对LLM进行了微调，使用pcap文件（标记为DDoS或良性）来分类流量。 Urban IoT是一个关于4060个物联网设备的真实世界匿名数据集。 由于该数据集的复杂性，他们根据是否考虑物联网设备之间流量的相关性来不同训练LLM。Suhaima J等人[177]提出了一个名为Improved Phishing and Spam Detection Mode*l*（IPSDM）的模型，这是Disti*l*BERT和RoBERTA的微调版本。该论文强调了LLMs改革电子邮件安全领域的潜力，并建议这些模型可以成为改善信息系统安全的有价值工具。 另一项工作也使用LLM进行垃圾邮件检测。Yuwei W等人[178]评估了ChatGPT在垃圾邮件检测中的性能，发现其在低资源中文数据集上优于BERT，但在更大的英文数据集上落后。 该研究还强调了增加提示示例对ChatGPT准确性的积极影响。Danie*l* N等人[179]引入了一种利用大型语言模型生成“提示上下文文档向量”的钓鱼邮件检测方法。通过向LLMs提出有针对性的问题关于电子邮件内容，该方法量化了常见说服原则的存在，创建捕捉钓鱼邮件中恶意意图的向量。 该技术利用了LLMs的推理能力，优于传统的钓鱼邮件检测方法。 除了检测钓鱼邮件，还尝试使用LLM生成钓鱼邮件。Fredrik H等人[180]评估了GPT-4在创建钓鱼邮件方面的性能，并将其效果与依赖于基于一般规则和认知启发式（V-Triad方法）的传统钓鱼方法进行了比较。 该论文还探讨了LLMs在检测钓鱼邮件中的应用，像GPT、C*l*aude、PaLM和LLaMA等模型展示了在识别恶意意图方面的强大能力，有时超过了人类的检测率。Noah Z等人[181]进行了一项关于网络入侵检测（NID）系统中决策树模型解释的研究。 他们将决策树的路径和结构数据转换为文本格式，并提供给LLM生成解释。 此外，LLM提供了额外的背景知识，帮助用户理解为什么某些特征在分类中很重要。Mohamed A F等人[182]通过采用一种称为Privacy-Preserving Fixed-Length Encoding（PPFLE）的新编码技术对网络流量进行编码。 然后，他们用这些编码数据训练了一个名为SecurityBERT的模型，执行网络流量的分类任务。 具体来说，他们的模型针对物联网设备，实现了在资源受限的物联网设备上高效准确的网络威胁检测。

Tarek A等人 [183]介绍了Hun*t*GPT，这是一个将大型语言模型与传统机器学习相结合用于网络异常检测的系统。 该系统利用在KDD99数据集上训练的随机森林分类器来识别网络威胁。 为了增强可解释性，它采用了SHAP和Lim*e*等XAI技术，并将它们与GPT-3.5对话代理结合使用。

数字取证。

Mark S等人 [184]评估了Ch*at*GPT在数字取证中的适用性。 Ch*at*GPT被用于帮助确定文件是否正在下载到个人电脑，并确定文件是否被特定用户执行。 此外，Ch*at*GPT还用于检测浏览器历史记录、Windows事件日志和与云平台机器的交互。

4.7 LLM辅助攻击

值得注意的是，虽然LLMs在提高网络安全性方面具有重要潜力，但它们也带来了新的挑战，涉及数据隐私、对抗性攻击以及需要持续培训和更新以跟上不断演变的威胁。 在2024年1月1日由Google组织的研讨会报告[185]中，双重使用问题

生成人工智能（GenAI）技术的重点被强调。 它们能够用于积极目的，也可能被用于恶意攻击。 在本节中，我们将详细讨论利用LLM发动的当前网络攻击。

LLM辅助攻击的当前状态。

Pawankumar S等人[186]指出Cha*r*GPT对网络安全既有积极影响，也可能产生负面影响。
在这篇文章中，他们列举了当前网络安全面临的各种威胁，包括恶意软件攻击、钓鱼、密码攻击等，并提到了ChatGPT在社会工程攻击中的潜在应用。Maanak G等人[187]也对生成人工智能在网络安全和隐私方面的影响进行了类似的研究。 此外，Stephen M等人[188]探讨了大型语言模型在网络威胁测试中的潜力，特别是在支持与威胁相关的行动和决策方面。 以虚拟机为例，他们详细讨论了如何对网络中的设备发动LLM指导的自动攻击。 作者得出结论，尽管这项工作还处于初步阶段，但它表明LLM在网络威胁方面具有强大潜力。 对于当前存在和可访问的恶意LLM，Zilong L等人[189]对212个真实世界的Malla（恶意LLM）进行了系统研究，揭示了它们如何在地下市场中传播和运作。 他们对Malla生态系统、开发框架、利用技术以及Malla在生成各种恶意内容方面的有效性进行了详细检查。 他们还提供了洞察力，了解网络犯罪分子如何利用LLM以及打击这种网络犯罪的策略。

利用LLM发起恶意攻击的手段。

LLM辅助特权升级攻击。**Andreas** H等人 [190]使用**LLM**来协助完成渗透测试。 他们开发了一个自动化的**Li**nux特权升级基准测试，以评估不同**LLM**的性能，并设计了一个名为W**inter**mu**te**的工具，用于快速探索**LL**M在引导特权升级方面的能力。
LLM辅助**CTF**（夺旗）挑战。W**esle**y **T**等人 [191]研究了现有大型语言模型在解决夺旗比赛挑战中的潜力。 他们使用三个已建立的模型（GP**T-**3.5、P**aLM**2和P**rometheus**），精选了常见**CTF**类别中具有代表性的挑战。
随后，进行了分析以评估LLM在解决这些挑战方面的表现。 他们的研究结果表明，LLM确实具有一定程度上协助参与者解决CTF挑战的能力，尽管并非全面。LLM辅助钓鱼网站/电子邮件生成。 Nils B等人 [192]使用LLM自动化生成高级钓鱼攻击。 在提出的攻击方法中，LLM用于以下功能：克隆目标网站、修改登录表单以捕获凭据、混淆代码、自动化域名注册和自动化脚本部署。Sayak S R等人 [193]检查了LLM（如Chat**GP**T、**GP**T-4、Claude和Bard）生成钓鱼攻击的潜力。 研究表明，这些模型可以有效地创建令人信服的钓鱼网站和电子邮件，模仿知名品牌，并采用规避策略以避免被检测。 该研究还开发了一种基于B**E**RT的检测工具，能够高准确度地识别恶意提示，作为对LLM被用于钓鱼诈骗的反制措施。

LLM辅助负载生成。 [194]尝试使用**LLM**的帮助编写负载以发动网络攻击。这项研究系统地探索了Ch**atGP**T生成2022年观察到的前10个**M**ITRE弱点的可执行代码，并将它们的性能与**Google**的B**ard**进行了比较。除了效率外，**LLM**生成的负载往往比手工制作的更复杂和有针对性。**LLM**启用的自动化渗透测试。**Gelei** D等人 [195]提出了一个名为**PentestGP**T的工具，旨在进行自动化渗透测试。 在**PentestGP**T中，进行了三个程序来执行渗透测试，包括推理、生成和解析模块。每个模块反映了渗透测试团队内的特定角色，以便系统能够尽可能模拟自动化渗透测试。**Andreas** H等人 [196]也进行了一项关于**LLM**帮助下的渗透测试的研究。该研究调查了两个用例：用于安全测试的高级任务规划和在易受攻击的虚拟机中进行低级漏洞搜索。 作者在**LLM**生成的操作和虚拟机之间实现了一个反馈循环，允许**LLM**分析系统状态以发现漏洞并提出攻击向量。**Jiacen** X等人 [197]介绍了**A**UTO**A**TT**A**CKER，这是一个利用**LLM**自动化"键盘操作"网络攻击的系统，模拟人类操作阶段。 该系统利用**LLM**生成各种技术和环境的精确攻击命令，将潜在的手动操作转化为自动化、高效的过程。 **A**UTO**A**TT**A**CKER由与**LLM**迭代交互的模块组成，利用总结、规划和行动选择等能力来构建复杂的攻击序列。攻击的代理。**Mika** B等人 [198]使用Ch**atGP**T作为受害者和攻击者（C&C）控制的网络之间的代理，这种模式允许他们远程控制受害者的系统而不直接通信，使追踪攻击者变得困难。

4.8 其他

除了先前描述的类别外，在网络安全领域还有一些零星研究关于LLM的应用，这些研究也具有研究价值。

物联网指纹。Armin S等人 [199]提出了一种用于生成互联网设备指纹的方法。 他们的方法分为两个步骤。首先，从网络扫描获得的原始文本数据将使用大型语言模型RoBERTa转换为稳定表示（嵌入）。接下来，使用HDBSCAN算法对嵌入代码进行聚类，进行下游任务。 因此，根据聚类生成指纹。

僵尸网络。Kaicheng Y等人 [200]阐明了一个名为fox8的由LLM驱动的Twitter僵尸网络。fox8僵尸网络包含超过一千名由人工智能控制的用户。他们发布机器生成的内容和窃取的图像，传播虚假和有害信息，通过回复和转发相互交流。

安全补丁检测。Xunzhu T等人提出了一个名为LLMDA的系统，其主要目标是改善开源软件（OSS）中安全补丁的识别。 LLMs用于生成补丁的解释性描述和合成数据，有助于增强现有数据集。[201]

SoC安全。Dipayan S等人 [202] 探讨了将大型语言模型（LLMs）整合到片上系统（SoC）安全验证范式中的潜力。 他们主要评估LLM在以下领域的应用：漏洞插入、安全评估、安全验证和对策开发。

污点分析。Puzhuo L等人 [203] 提出了一种名为LATTE的静态二进制污点分析工具，由LLMs支持。LLM有助于识别污点来源和可能的漏洞触发器之间的数据依赖链（危险流）。 LLM在过程中提供了对代码结构和语义的理解。

LLMs的输入输出保护。 [204]设计了一个名为Llama Guard的模型，主要专注于检测LLM提示/响应中的风险。 基于Llama2-7b模型，对收集到的标记有安全风险的文本进行指令调整。

蜜罐。Muris S等人 [205]设计了一个动态实时的虚假蜜罐，通过LLM生成的响应来改变蜜罐容易被识别的局限性。 在他们的实验中，大多数人无法辨别远程主机是真实的还是LLM生成的蜜罐。

事件响应。Sam H和Jules W [206]主张在网络安全中应用ChatGPT来增强事件响应规划（IRP）。 它表明LLMs可以起草初步计划，推荐最佳实践，并识别文档漏洞。 该论文强调了LLMs简化IRP流程的潜力，强调人类监督的价值，以确保准确性和相关性。

网络管理。 [207]探讨了如何利用LLM从自然语言查询中生成特定任务代码，以改进网络管理。 他们开发并发布了一个测试基准，NeMoEval，涵盖了两个网络管理应用程序：网络流量分析和网络生命周期管理。

漏洞再现。斯东・F等人。 [208]提出了一种名为AdbGPT的方法，利用大型语言模型从漏洞报告中自动复制漏洞，通过提示工程师进行工程而无需培训或硬编码。

关于网络安全领域的专业问答。萨米亚・K等人。 [209]是对ChatGPT在回答Stack Overflow编程问题中表现的实证研究。 LLM响应的主要缺点在于虚假信息和内容过长。 尽管一些测试人员喜欢其全面性和良好的语言表达风格。 由于难以识别LLM提供的误导信息，这是一个尚未研究的领域。

回答问题2：LLM在网络安全领域展现出巨大潜力，在威胁情报、异常检测、漏洞检测等各个方面提供帮助。LLM安全副驾驶可以有效增强网络安全的自动化和智能化，有助于解决安全风险挑战。 尽管相关研究取得了一定进展，但在更好地应用LLM在网络安全领域方面仍值得进一步探索。

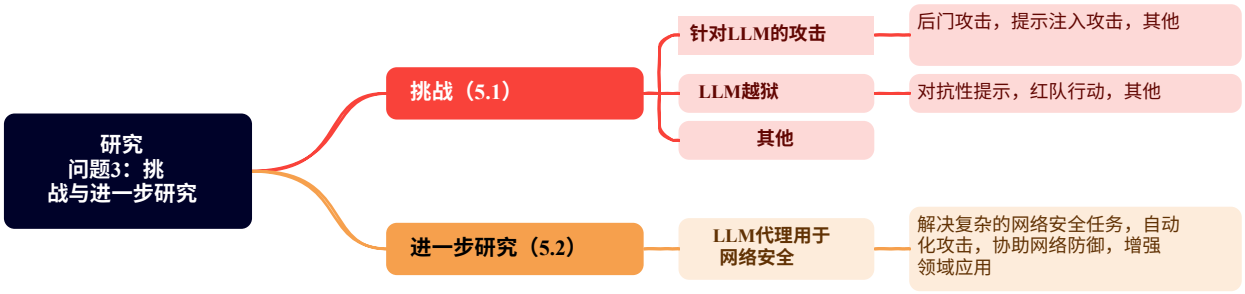


图7： 研究问题3的概述。

研究问题3：关于LLM在网络安全中应用的现有挑战和进一步研究方向是什么？

5.1 挑战

在网络安全领域应用LLMs代表着一个前沿的切口，展示了LLMs在解决复杂和动态网络威胁方面的强大能力。 然而，尽管它们有优势，LLMs并非毫无挑战，特别是在固有的漏洞和易受攻击方面[16, 210]。 LLMs导向攻击和LLMs越狱现象是关键关注点之一。 这些漏洞突显了LLMs在网络安全中应用的双刃性质。 一方面，LLMs强大的理解和预测能力可以显着提升网络安全系统的智能。 另一方面，它们固有的弱点为利用提供了途径，带来严重的安全风险，削弱了它们在网络安全应用中的可靠性和完整性。

我们从两个关键角度探讨这些挑战：针对LLMs的攻击，检验LLMs对各种形式攻击的易受性[211, 212, 213]，以及LLMs越狱，关注LLMs在某些方式下生成不安全或意外内容的现象，尽管设计时带有保障[214, 215]。 通过对这些维度的分析，我们旨在阐明在网络安全中利用LLMs的复杂性，强调在应用中需要谨慎和战略远见。

针对LLMs的攻击。

LLMs的漏洞使它们容易受到恶意用户的攻击。 我们关注两种类型的攻击：后门攻击和提示注入攻击。

后门攻击通过在模型或其输入中嵌入特定触发器来操纵模型输出，以实现攻击者的目标。Jiawen S等人[216]提出了一种新颖的后门攻击方法，称为BadGPT，专门针对通过强化学习进行微调的语言模型，如ChatGPT。该方法涉及在奖励模型中嵌入后门，可以通过特定触发提示激活。 这种激活允许攻击者扭曲模型的输出以符合其偏好，展示了一个关键的安全漏洞。 在另一项研究中，Shuai Z等人[217]介绍了一种新颖的后门攻击策略ICLAttack，旨在利用LLMs的固有上下文学习能力。ICLAttack框架包括两个主要攻击向量：毒化演示示例和毒化演示提示。 通过在模型的上下文中嵌入后门触发器，ICLAttack能够影响模型的行为而无需微调，从而揭示了LLMs内部的普遍漏洞。此外，Hongwei Y等人[218]揭示了一种针对基于提示的LLMs量身定制的后门攻击机制，称为PoisonPrompt。该方法通过两个步骤将后门注入语言模型：毒化提示生成和双层优化。 这种方法可以在特定触发器激活下改变模型的正常预测，而不影响模型在下游任务上的性能，对LLMs的完整性构成微妙但强大的威胁。

提示注入攻击涉及攻击者将恶意命令插入输入中，迫使模型执行与攻击者意图一致的操作。Rodrigo P等人[219]对基于Langchain框架的Web应用程序针对提示到SQL（P2SQL）注入攻击进行了全面调查。 这些攻击利用用户输入提示生成恶意SQL查询，从而使攻击者能够篡改数据库或窃取敏感信息。Shuyu J等人[220]介绍了组合指令攻击（CIA），揭示了LLMs对利用具有潜在恶意意图的组合指令进行攻击的易受攻击性。 通过Talking-CIA和Writing-CIA两种转换方法，有害指令被伪装成对话或写作任务，使模型无法辨别潜在的恶意意图，从而生成有害内容。Yupei

L等人[221]提出了一种名为HOUYI的新型黑盒提示注入攻击技术，用于集成LLMs的应用程序。HOUYI通过三个关键元素执行攻击：预构建提示、注入提示和恶意负载。 在36个真实场景中的部署凸显了其在揭示和利用LLM集成应用程序中的漏洞方面的有效性。等人[222]专注于针对指令调整的LLMs的虚拟提示注入（VPI）攻击，允许攻击者通过指定虚拟提示来操纵模型行为，而无需直接注入模型输入，导致模型传播偏见信息。 J*at*mo [223]利用指令调整模型生成特定任务的数据集。 然后利用这些数据集对基础模型进行微调，增强其抵抗大多数提示注入攻击的鲁棒性。

此外，George K等人 [224] 以半自动化方式构建了名为AttaQ的对抗攻击数据集，旨在评估LLMs在面对有害或不当输入时的安全性。 通过分析模型对AttaQ数据集的响应，暴露了漏洞，并进一步应用专门的聚类技术来识别和描述模型的易受攻击的语义区域。Aysan E等人 [212] 对针对LLMs的各种攻击类型进行了全面调查，包括直接对模型本身的攻击和间接对利用模型的应用程序的攻击。 本研究描述了这些攻击对模型的隐私、安全性和可靠性的影响。 并强调在开发AI模型时实施积极的安全措施的重要性。

LLMs越狱。

如上所述，LLM容易受到各种攻击，其中越狱攻击是最流行的之一。
辛悦 S等人 [225] 调查了LLMs面对越狱提示时的安全问题，收集并分析了6,387个提示，揭示了这些提示的特征和攻击策略。 尽管LLMs实施了各种安全措施，但他们发现有效的越狱提示仍然成功地诱使模型生成有害内容，表明LLMs在安全方面需要进一步改进。君杰 C等人 [214] 对LLMs越狱进行了全面评估，揭示了这些攻击方法的有效性以及LLMs在各种违规类别中的漏洞。

有各种方法可以生成对抗性提示。安迪 Z等人 [226] 结合贪婪搜索和基于梯度的优化技术，提出了一种方法，可以自动生成对抗性后缀以提示模型，包括开源和商业模型，以生成不当内容。拉兹 L等人 [227] 引入了一种使用遗传算法进行黑盒越狱攻击的新方法，可以操纵LLMs生成意外和潜在有害输出，而无需访问模型的内部结构和参数，通过优化通用对抗性提示。彭 D等人 [228] 将越狱过程概念化为提示重写和场景嵌套。 然后，他们引入了一个越狱提示生成框架ReNeLLM，利用LLMs自身生成有效的越狱提示。 ReNeLLM在多个LLMs上实现了高攻击成功率，同时与现有基线相比显着降低了时间成本。格雷 D等人 [229] 探索了LLM Chatbots上的越狱攻击，并提出了一个名为MASTERKEY的框架来自动化这一过程。 通过时间特征分析和自动提示生成，MASTERKEY揭示并绕过LLM chatbots的防御机制，为LLM安全研究提供新视角，并为服务提供商改进安全措施提供指导。

LLMs越狱研究也可以用于红队。Sicheng Z等人 [230]提出了AutoDAN，一种可解释的、基于梯度的对抗攻击方法。 通过将越狱和可读性的双重目标结合起来，它生成了可解释和多样化的攻击提示，能够有效地规避困惑过滤器，并在训练数据有限的情况下展示了强大的泛化能力。 这种方法不仅为LLMs的红队提供了一种新颖的方法，还有助于理解越狱攻击的机制。Jiahao Y等人 [231]提出了一个名为GPTFUZZER的新的黑盒越狱模糊测试框架。通过从互联网收集人类编写的越狱模板作为初始种子，然后通过种子选择、突变和评估攻击成功的过程迭代，GPTFUZZER显着提高了红队测试的效率和可扩展性。Dongyu Y等人 [232]介绍了FuzzLLM，一种新颖且普遍适用的模糊测试框架，旨在主动发现LLMs中的越狱漏洞。利用基于模板的策略，FuzzLLM可以生成各种越狱提示，并通过自动化测试识别潜在的安全漏洞。 它在各种LLMs上展示了高效性和全面性，有效地识别和评估越狱漏洞。

此外，Zhenhua W等人 [233] 提出了语义防火墙的概念，用来描述LLMs对恶意提示的防御机制，并提出了一种自欺攻击方法来绕过LLM语义防火墙。
该方法涉及设计一个可定制的对话模板，用于实验特定非法有效载荷，并自动实现LLM越狱。Huachuan Q等人 [234] 开发了一个潜在越狱提示数据集，嵌入恶意指令，并使用分层注释框架对LLM在不同条件下的性能进行系统分析，如指令位置、词替换和指令替换。这旨在评估LLMs在处理包含潜在恶意指令的文本时的安全性和输出稳健性。

Haoran L等人 [235] 调查了与ChatGPT和集成了ChatGPT的Bing搜索引擎相关的潜在隐私威胁。通过引入一种新颖的多步越狱提示，他们成功从ChatGPT中提取个人可识别信息，并展示了新Bing在直接提示下带来的隐私威胁。

其他人。

除了广泛研究的漏洞外，还有一些其他LLM风险限制了它们在网络安全中的应用。
Pasupuleti R等人[236]强调了生成式人工智能和ChatGPT的双刃性质，揭示了它们带来的诸多网络安全和伦理挑战，以及它们的便利性。Evan H等人[237]研究了LLM在某些触发条件下可能表现出的欺骗行为，并发现即使在安全训练后，这些行为可能仍然存在，对人工智能系统的安全构成潜在威胁。Xianjun Y等人[238]指出，即使是安全对齐的LLM也可以很容易地被操纵以生成有害内容，突显了维护LLM安全的复杂性。Fengqing J等人[239]在LLM集成应用程序中确定了关键漏洞，这些漏洞可能源自恶意应用程序开发人员或具有控制数据库访问、操纵和污染数据能力的外部威胁。Sallou J等人[240]还对与使用闭源LLM相关的数据泄露和可重现性问题提出了担忧。

回答问题3：尽管LLM具有强大的能力，但它们固有地具有某些弱点和漏洞，使它们容易受到攻击。特别是越狱对LLM应用造成重大安全风险。

5.2 进一步研究

尽管在网络安全领域对LLM进行了重要研究，但对这些模型的探索和应用仍处于初级阶段，具有巨大的增长潜力[18, 19]。 网络安全的复杂性不仅源于攻击方法的多样性，还源于网络环境的复杂性，加上需要全面应用各种工具和策略以实现有效保护的需求[241,242]。 面对这些挑战，需要具有增强规划、推理、工具使用、记忆等能力的AI系统。因此，LLM代理的概念应运而生，引起研究人员的广泛关注。

LLM代理是"一个可以使用LLM来推理问题、创建解决问题的计划，并在一组工具的帮助下执行计划的系统"[243]。 通过模拟复杂的网络行为和攻击模式，并整合先进的自然语言处理能力，LLM代理为网络安全领域引入了新的视角和解决方案[96, 188, 244, 245, 246, 247]。 随着技术的不断进步和更深入的研究，LLM代理有望在制定防御策略、威胁检测和制定安全政策方面发挥关键作用，显着提高网络安全防御的效率和智能水平。

基于LLMs的AI代理框架具有解决复杂问题所需的关键能力 [248]。
Zhiheng X等人 [249] 提出了一个包括大脑、感知和行动组件的LLM代理架构，提供了在单一代理场景、多代理环境和人-代理协作中广泛应用的可能性。
此外，工具和API调用的整合赋予LLM代理与现实世界互动的能力。 ToolLLM [250] 开发了ToolBench数据集和DFSDT算法，使LLMs能够成功处理涉及众多现实世界API的复杂任务。Sum2Act [251] 通过总结结果并做出明智决策，引入了一种复杂的工具调用机制，增强了LLMs与外部工具的互动。

此外，Ke Y等人 [252] 表明将代码（编程语言）整合到LLMs中显着增强了它们的能力，使它们能够承担更复杂的智能代理任务。Bo Q等人 [253] 提出了TaskWeaver，一个以代码为先的代理框架，用于无缝规划和执行数据分析任务。

LLM代理可以应用于解决复杂的网络安全任务。 LLMind [244] 是一个创新框架，利用LLMs作为协调员，与物联网设备和领域特定的人工智能模块集成，执行复杂任务。 该框架采用有限状态机方法生成控制脚本，从而提高任务执行的准确性和成功率。 此外，LLMind引入了一个积累经验的机制，允许系统通过用户和机器之间的持续互动不断学习和进步。

Maria R等人[245]展示了LLMs在网络安全环境中作为代理的应用。 在N*et*S*e*cGam*e*和Cyb*e*rBa*ttle*Sim设置中的实验表明，LLM代理在顺序决策任务中可以达到与经过广泛训练的代理相媲美甚至更好的性能，即使没有额外的训练。 此外，该研究介绍了N*et*S*e*cGam*e*环境，这是一个高度模块化和适应性强的网络安全环境，旨在支持复杂的多代理场景。Yudong H等人[254]提出了Cha*t*N*et*，一个领域特定的网络LLM框架

具有访问各种外部网络工具的权限。 ChatNet显着减少了繁琐的网络规划任务所需的时间，从而大幅提高了效率。

LLM代理可以用于执行自动攻击。Richard F等人[246]揭示了LLM代理在网络安全攻击中的潜力，特别是GPT-4在没有先验漏洞知识的情况下自主进行复杂的黑客攻击的能力。 研究表明，LLM代理在黑客尝试中的成功率高达73.3%，并且可以自主发现现实世界网站中的漏洞。Stephen M等人。

[188]展示了LLM在网络威胁测试中的潜在应用，特别是在自动化网络攻击活动方面。 通过及时工程和设计自动代理，LLM可以理解并执行复杂的网络攻击任务。

LLM代理还可以用于协助网络防御。 Nissist [247]被设计为一个多代理系统，精确理解用户查询并提供有效的缓解计划。 Nissist利用故障排除指南和事件缓解历史提供积极的建议，大大减少了事件缓解的时间，减轻了值班工程师的工作量，并增强了服务的可靠性。 Cyber Sentinel [96]是基于GPT-4的对话代理，可以解释潜在的网络威胁并根据用户指令执行安全操作。 Cyber Sentinel在网络安全运营中的潜在影响包括提高的威胁检测和响应能力，增强的运营效率，实时协作和知识共享。

LLM代理通过其卓越的能力增强了网络安全应用，然而代理系统固有的安全风险[255]在网络安全环境中部署它们时带来挑战。方洲等人[256]提出了基于Web的间接提示注入的概念，这是一种新型的网络威胁，它在网页中嵌入恶意指令以间接控制这些代理，实现了在不同用户输入下高成功率和鲁棒性。秋思等人[257]指出，LLM代理通过集成外部工具可能导致间接提示注入攻击的风险，攻击者在LLM处理的内容中嵌入恶意命令，操纵这些代理执行对用户有害的操作。

总之，LLM代理在网络安全中的应用为解决数字安全威胁开辟了新途径。 尽管这一领域的研究仍处于早期阶段，并且代理固有的安全漏洞尚未得到解决，但这一研究方向有望显着增强对复杂网络威胁的应对能力，并有潜力彻底改变安全专业人员的工作方法，从而释放更大的生产力。 因此，进一步研究LLM代理在网络安全中的应用对于开发适应性强、智能化和全面的网络安全解决方案至关重要。

> 回答问题3：扩展LLM的工具使用和API调用能力，结合设计能够理解、规划决策并在网络安全应用中执行复杂任务的自主智能代理，将显着推动人工智能在网络安全领域的利用。

6 结论

本文阐述了构建面向网络安全领域LLM的方法论，详细说明了如何利用有针对性的数据对现有模型进行微调以满足特定需求。 对LLM的潜在应用进行的调查揭示了它们在许多网络安全任务中的巨大潜力，如威胁情报、漏洞检测、安全代码生成等。 然而，我们也意识到LLM的固有漏洞，特别是它们容易受到越狱等攻击的影响，这会带来重大安全风险。 减轻这些漏洞对于在敏感环境中安全部署LLM至关重要。 此外，我们提出未来的研究方向，如扩展LLM的工具使用和API调用能力，以及为复杂网络安全操作开发自主智能代理。

总的来说，我们弥合了LLM的进展与网络安全需求之间的差距，为研究人员和实践者奠定了基础。 它指导他们利用LLM的变革潜力，同时解决在这一领域中出现的独特挑战。 在这个方向上的持续努力对于定义网络安全的未来以及与LLM卓越能力的整合至关重要。

参考文献

[1] OpenAI ChatGPT. https://openai.com/chatgpt .

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, 等. Llama: 开放高效的基础语言模型.arXiv预印本 arXiv:2302.13971, 2023.

[3] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, 等. Vicuna: 一个开源聊天机器人，以90%*的chatgpt质量令人印象深刻。见 https://vicuna. lmsys. org (访问日期2023年4月14日), 2(3):6, 2023.

[4] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, 等。The falcon series of open language models.arXiv 预印本 arXiv:2311.16867, 2023.

[5] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, 等。Mixtral of experts. *arXiv* 预印本 *arXiv:2401.04088*, 2024.

[6] Barret Zoph, Colin Raffel, Dale Schuurmans, Dani Yogatama, Denny Zhou, Don Metzler, Ed H Chi, Jason Wei, Jeff Dean, Liam B Fedus, 等。Emergent abilities of large language models. *TMLR*, 2022

[7] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 大型语言模型：一项调查。arXiv预印本 arXiv:2402.06196, 2024.

[8] Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, 等。Openagi：当llm遇见领域专家。神经信息处理系统的进展, 36, 2024.

[9] Pravneet Kaur, Gautam Siddharth Kashyap, Ankit Kumar, Md Tabrez Nafis, Sandeep Kumar, 和 Vikrant Shokeen。从文本到转换：大型语言模型多功能性的全面审查。 *arXiv*预印本 arXiv:2402.16142, 2024.

[10] 侯新艺，赵燕洁，刘悦，杨洲，王凯龙，李力，罗霞普，David Lo，John Grundy和王浩宇。用于软件工程的大型语言模型：系统文献综述。arXiv预印本*arXiv:2308.10620*，2023年。

[11] 赖金琦，甘文胜，吴家阳，齐振莲和Philip S. Yu。法律领域的大型语言模型：一项调查。*arXiv*预印本*arXiv:2312.03718*，2023年。

[12] 周宏健，刘丰林，顾博洋，邹新宇，黄金发，吴晶娥，李怡茹，Sam S. Chen，周培林，刘俊玲，华一宁，毛成峰，游晨宇，吴贤，郑业峰，Lei Clifton，李政，罗杰波和David A. Clifton。医学领域的大型语言模型综述：进展、应用和挑战。
*arXiv*预印本 *arXiv:2311.05112*, 2024.

[13] Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 大型语言模型在教育中的实际和伦理挑战: 一项系统范围审查。英国教育技术杂志, 55(1):90–112, 2024.

[14] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 金融中的大型语言模型: 一项调查。 在第四届ACM国际金融人工智能会议论文集中, 页码374–382, 2023.

[15] Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, 等. Chemdfm: 化学对话基础模型。arXiv预印本 arXiv:2401.14818, 2024.

[16] 姚一凡，段金豪，徐凯迪，蔡元芳，孙恩，张悦。关于大型语言模型（LLM）安全和隐私的调查：好的，坏的和丑陋的。arXiv预印本 arXiv:2312.02003，2023年。

[17] 达斯巴德汉・钱德拉，阿米尼・M・哈迪和吴彦照。大型语言模型的安全和隐私挑战：一项调查。arXiv预印本 arXiv:2402.00888，2024年。

[18] 加布里埃尔・德・杰苏斯・科埃略・达・席尔瓦和卡洛斯・贝克尔・韦斯特法尔。网络安全中大型语言模型的调查。*arXiv*预印本 *arXiv:2402.16968*，2024年。

[19] 法扎德・努尔莫哈迪扎德・莫特拉格，梅尔达德・哈吉扎德，梅赫里亚尔・马吉德，佩吉曼・纳贾菲，冯成，克里斯托夫・迈内尔。网络安全中的大型语言模型：最新技术。arXiv预印本 arXiv:2402.00891，2024年。

[20] Yagmur Yigit, William J Buchanan, Madjid G Tehrani, and Leandros Maglaras. 安全领域中生成式人工智能方法综述。arXiv预印本 arXiv:2403.08701, 2024.

[21] Kutub Thakur, Meikang Qiu, Keke Gai, and Md Liakat Ali. 关于网络安全威胁和安全模型的调查。 在2015年IEEE第2届国际网络安全与云计算会议, 页码307–311。IEEE, 2015.

[22] Natalie M Scala, Allison C Reilly, Paul L Goethals, and Michel Cukier. 风险与网络安全的五大难题。风险分析, 39(10):2119–2126, 2019.

[23] Diptiben Ghelani. 网络安全、网络威胁、影响和未来展望：一篇综述。 *Authorea* 预印本, 2022.

[24] Yuchong Li 和 Qinghui Liu。关于网络攻击和网络安全的综合审查研究；新兴趋势和最新发展。能源报告，7:8176–8186，2021年。

[25] Ömer Aslan，Semih Serkant Aktu˘g，Merve Ozkan-Okay，Abdullah Asim Yilmaz 和 Erdal Akin。网络安全漏洞、威胁、攻击和解决方案的综合审查。 电子，12(6):1333，2023年。

[26] Pranav Kumar Chaudhary。人工智能、机器学习和大型语言模型在网络安全中的应用。

[27] Hugo Touvron，Louis Martin，Kevin Stone，Peter Albert，Amjad Almahairi，Yasmine Babaei，Nikolay Bashlykov，Soumya Batra，Prajjwal Bhargava，Shruti Bhosale，等。Llama 2：开放基础和精细调整的聊天模型。*arXiv* 预印本 *arXiv:2307.09288*, 2023.

[28] André Silva, Sen Fang, 和 Martin Monperrus. Repairllama: 用于程序修复的高效表示和微调适配器。ar*X*iv预印本 ar*X*iv:2*31*2.*1569*8, 2023.

[29] Jie Zhang, Hui Wen, Liting Deng, Mingfeng Xin, Zhi Li, Lun Li, Hongsong Zhu, 和 Limin Sun. Hackmentor: 为网络安全调整大型语言模型。 在2*0*2*3*年I*EEE*国际信任、安全和隐私计算与通信会议（*T*rust*C*om）. IEEE, 2023.

[30] Shafi Parvez Mohammed 和 Gahangir Hossain. Chatgpt在教育、医疗保健和网络安全中的机遇和挑战。 在*20*2*4*年I*EEE*第*14*届年度计算与通信研讨会和会议（CC*W*C）, 页码03*1*6–03*21*. I*EEE*, *20*2*4*.

[31] Rahul Pankajakshan, Sumitra Biswal, Yuvaraj Govindarajulu, and Gilad Gressel. 映射LLM安全领域：一项全面的利益相关者风险评估提案。ar*X*iv预印本ar*X*iv:*240*3.133*09*, 2024.

[32] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, 等。语言模型是少样本学习者。神经信息处理系统的进展, 33:1877–1901, 2020.

[33] Gemini团队, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, 等。Gemini：一系列高度能干的多模型。ar*X*i*v*预印本ar*X*i*v*:23*12*.*11805*, 2023.

[34] Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, 等。Code llama: 用于代码的开放基础模型。arXi*v*预印本*arXiv:2308.12950*, 2023.

[35] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, 等。Starcoder: 愿源码与你同在！ar*X*iv预印本 ar*X*iv:23*0*5.*06161*, 2023.

[36] Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, 等。Starcoder 2和stack v2: 下一代。 *arXiv* 预印本 *arXiv:2402.19173*, 2024.

[37] Ramanpreet Kaur, Dušan Gabrijelˇciˇc, and Tomaž Klobuˇcar. 人工智能在网络安全中的应用：文献综述和未来研究方向。信息融合，第101804页，2023年。

[38] Sarvesh Kumar, Upasana Gupta, Arvind Kumar Singh, and Avadh Kishore Singh. 人工智能：改变数字时代的网络安全。计算机、机械和管理杂志，2(3)：31–42，2023年。

[39] Maad Mijwil, Mohammad Aljanabi等。走向基于人工智能的网络安全：实践和*C*hatGPT生成的对抗网络犯罪的方法。 伊拉克计算机科学与数学杂志，4(1)：65–70，2023年。

[40] Ça˘gatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. 探究大型语言模型中的持续预训练：见解和影响。 ar*X*iv预印本 ar*X*iv:*2*402.*17*400，*20*24年。

[41] 张铁铮，陈晓曦，屈崇宇，Alan Yuille和周宗伟。在交互式分割中利用AI预测和专家修订的注释：持续调整还是完全训练？ arXiv预印本*arXiv:2402.19423*，2024年。

[42] 张胜宇，董林峰，李晓亚，张森，孙晓飞，王树和，李继伟，胡润一，张天威，吴飞等。大型语言模型的指令调优：一项调查。arXiv预印本arXiv:2308.10792，2023年。

[43] 董冠廷，袁宏毅，陆克明，李成鹏，薛明峰，刘大毅，王伟，袁政，周畅和周靖仁。大型语言模型的能力如何受到监督微调数据组成的影响。arXiv预印本arXiv:2310.05492，2023年。

[44] 宁丁，秦宇佳，杨光，魏福超，杨宗瀚，苏玉生，胡胜鼎，陈玉林，陈启民，陈维泽等。大规模预训练语言模型的参数高效微调。
自然机器智能，5(3)：220–235，2023年。

[45] 诺伯特・提哈尼，穆罕默德・阿明・费拉格，里迪・贾因和梅鲁安・德巴。Cybermetric：用于评估大型语言模型在网络安全领域知识的基准数据集。arXiv预印本 arXiv:2402.07688，2024年。

[46] 曼尼什・巴特，萨哈娜・切纳巴萨帕，赛勒斯・尼古拉伊迪斯，万胜业，伊万・埃夫蒂莫夫，多米尼克・加比，丹尼尔・宋，法伊赞・艾哈迈德，康奈利厄斯・阿施曼，洛伦佐・方塔纳，萨沙・弗罗洛夫，拉维・普拉卡什・吉里，达瓦尔・卡皮尔，伊安尼斯・科兹拉基斯，大卫・勒布朗，詹姆斯・米拉佐，亚历山大・斯特劳曼，加布里埃尔・辛纳夫，瓦伦・沃恩蒂米塔，斯宾塞・惠特曼和约书亚・萨克斯。紫色羊驼网络安全评估：语言模型的安全编码基准。arXiv预印本 arXiv:2312.04724，2023年

[47] Catherine Tony, Markus Mutas, Nicolás E. Díaz Ferreyra, and Riccardo Scandariato. Llmseceval: 用于安全评估的自然语言提示数据集。arXiv预印本 arXiv:2303.09384, 2023.

[48] Mohamed Amine Ferrag, Ammar Battah, Norbert Tihanyi, Merouane Debbah, Thierry Lestable, and Lucas C Cordeiro. Securefalcon: 下一代网络安全推理系统。arXiv预印本 arXiv:2307.06616,2023.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 注意力就是一切。神经信息处理系统的进展, 30, 2017.

[50] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever等。通过生成式预训练改善语言理解。2018年。

[51] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan等。Baichuan 2: 开放式大规模语言模型。arXiv预印本 arXiv:2309.10305, 2023年。

[52] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu和Gholamreza Haffari。大型语言模型的持续学习: 一项调查。arXiv预印本 arXiv:2402.01364, 2024年。

[53] Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky和Irina Rish。持续预训练大型语言模型的简单且可扩展的策略。 arXiv预印本 arXiv:2403.08763, 2024年。

[54] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. 关于适配器调整对预训练语言模型适应性的有效性。arXiv预印本arXiv:2106.03164, 2021.

[55] 刘晓, 郑亚楠, 杜正晓, 丁明, 钱玉杰, 杨志林, 唐杰。Gpt也能理解。
AI Open, 2023.

[56] 刘晓, 季凯轩, 傅一成, 谭荣林, 杜正晓, 杨志林, 唐杰。P-tuning v2:提示调整可以在各种规模和任务上普遍与微调相媲美。 arXiv预印本arXiv:2110.07602, 2021.

[57] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv*预印本 *arXiv:2104.08691*, 2021.

[58] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models.arXiv预印本 arXiv:2106.09685, 2021.

[59] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[60] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities.arXiv预印本 arXiv:2305.13172,2023.

[61] 张宁宇，姚云志，田博忠，王鹏，邓树敏，王梦茹，奚泽坤，毛胜宇，张金天，倪元胜，程思源，徐子文，徐鑫，顾佳晨，蒋勇，谢鹏军，黄飞

，梁磊，张志强，朱晓伟，周军，陈华军。大型语言模型知识编辑的全面研究。arXiv预印本 arXiv:2401.01286，2024年。

[62] 阿拉斯·博兹库特和拉梅什·C·沙玛。生成式人工智能和提示工程：向算法世界中的精灵耳语的艺术。亚洲远程教育杂志，18(2)：i–vii，2023年。

[63] 叶钦源，马克萨姆德，里德普赖赞特，费雷什特卡尼。提示工程师的提示工程。 arXiv 预印本 arXiv:2311.05661，2023年。

[64] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 大型语言模型中提示工程的系统调查：技术和应用。arXiv预印本arXiv:2402.07927, 2024.

[65] 刘泽芳，史佳磊，和约翰F布福德。Cyberbench：用于评估大型语言模型在网络安全中的多任务基准。

[66] 李冠诚，李一峰，王冠男，杨浩宇，和余扬。Seceval：用于评估基础模型网络安全知识的全面基准。 https://github.com/XuanwuAI/SecEval, 2023.

[67] 刘泽芳。Secqa：用于评估大型语言模型在计算机安全中的简明问答数据集。arXiv预印本 arXiv:2312.15838, 2023.

[68] Mohammed Latif Siddiq 和 Joanna C. S. Santos. Securityeval 数据集: 挖掘漏洞示例以评估基于机器学习的代码生成技术。 在第1届国际软件存储库隐私和安全应用研讨会, MSR4P&S 2022, 页码 29–33, 美国纽约, 2022年。 计算机协会。

[69] Kamel Alrashedy 和 Abdullah Aljasser. LLMs 能够修补安全问题吗？ arXiv 预印本 arXiv:2312.00024, 2024年。

[70] 田润楚, 叶一宁, 覃宇佳, 丛鑫, 林彦凯, 刘知远 和 孙茂松. Debugbench: 评估大型语言模型的调试能力。arXiv 预印本 arXiv:2401.04621, 2024年。

[71] 苗宇凯，白宇，陈力，李丹，孙海峰，王希政，罗自秋，任燕宇，孙大鹏，徐秀婷，张琦，向超，和李新驰。预训练大型语言模型的网络运维能力的实证研究。arXiv预印本 arXiv:2309.05557，2023年。

[72] 刘宇和，裴长华，徐龙龙，陈博涵，孙明泽，张智睿，孙永谦，张胜林，王坤，张海明，李建辉，谢高刚，温希道，聂晓辉，马明华，和裴丹。Opseval：大型语言模型的全面IT运维基准套件。arXiv预印本arXiv:2310.07637，2024年。

[73] Alexey Shestov, Rodion Levichev, Ravil Mussabayev, and Anton Cheshkov. 为漏洞检测调整大型语言模型。arXiv预印本 arXiv:2401.17010, 2024.

[74] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: 用evol-instruct赋能代码大型语言模型。arXiv预印本 arXiv:2306.08568, 2023.

[75] Aidan ZH Yang, Claire Le Goues, Ruben Martins, and Vincent Hellendoorn. 用于无需测试的故障定位的大型语言模型。 在第46届IEEE/ACM国际软件工程会议论文集, 页码1–12, 2024.

[76] Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. Codegen: 一个用于代码的开放大型语言模型，具有多轮程序综合。arXiv预印本arXiv:2203.13474, 2023.

[77] Erik Nijkamp, Hiroaki Hayashi, Caiming Xiong, Silvio Savarese, and Yingbo Zhou. Codegen2: 训练LLMs处理编程和自然语言的经验教训。arXiv预印本 arXiv:2305.02309, 2023.

[78] André Storhaug, Jingyue Li, and Tianyuan Hu. 利用漏洞约束解码有效地避免智能合约代码中的漏洞。 在2023年IEEE第34届国际软件可靠性工程研讨会(ISSRE), 页码683–693. IEEE, 2023.

[79] Ben Wang. Mesh-Transformer-JAX: 使用JAX进行模型并行实现的Transformer语言模型。 https://github.com/kingoflolz/mesh-transformer-jax ，2021年5月。

[80] Jingxuan He, Mark Vero, Gabriela Krasnopolska, 和 Martin Vechev. 用于安全代码生成的指令调优。arXiv预印本 arXiv:2402.09497，2024年。

[81] Nan Jiang, Chengxiao Wang, Kevin Liu, Xiangzhe Xu, Lin Tan, 和 Xiangyu Zhang. Nova$^+$: 用于二进制的生成式语言模型。arXiv预印本 arXiv:2311.13721，2023年。

[82] Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, 等. Owl: 用于IT运维的大型语言模型。arXiv预印本 arXiv:2309.09298，2023年。

[83] Shaswata Mitra, Subash Neupane, Trisha Chakraborty, Sudip Mittal, Aritran Piplai, Manas Gaur, and Shahram Rahimi. Localintel: 从全球和本地网络知识生成组织威胁情报。arXiv预印本 arXiv:2401.10036, 2024.

[84] Filippo Perrina, Francesco Marchiori, Mauro Conti, and Nino Vincenzo Verde. Agir: 使用自然语言生成自动化网络威胁情报报告。arXiv 预印本 arXiv:2310.02655, 2023.

[85] Reza Fayyazi and Shanchieh Jay Yang. 利用大型语言模型解释模糊的网络攻击描述。arXiv 预印本 arXiv:2306.14062, 2023.

[86] Reza Fayyazi, Rozhina Taghdimi, and Shanchieh Jay Yang. 推进ttp分析：利用仅编码器和仅解码器语言模型的力量与检索增强生成。 arXiv 预印本arXiv:2401.00280, 2024.

[87] Tanmay Singla, Dharun Anandayuvaraj, Kelechi G. Kalu, Taylor R. Schorlemmer, and James C. Davis. 使用大型语言模型分析软件供应链安全失败的实证研究。 在2023年软件供应链攻击研究和生态系统防御研讨会论文集中, SCORED'23, 页码5–15, 美国纽约, 2023年。 计算机协会。

[88] Samaneh Shafee, Alysson Bessani, 和 Pedro M. Ferreira. 评估用于OSINT基础的网络威胁意识的LLM聊天机器人。arXiv预印本 arXiv:2401.15127, 2024年。

[89] Gaëtan Michelet 和 Frank Breitinger. ChatGPT, LLAMA, 你能帮我写报告吗？ 一项关于使用（本地）大型语言模型撰写辅助数字取证报告的实验。arXiv预印本 arXiv:2312.14607, 2023年。

[90] Giuseppe Siracusano, Davide Sanvito, Roberto Gonzalez, Manikantan Srinivasan, Sivakaman Kamatchi, Wataru Takahashi, Masaru Kawakita, Takahiro Kakumaru, and Roberto Bifulco. 时间到了：自动分析野外的网络威胁情报。arXiv预印本 arXiv:2307.10214, 2023.

[91] 胡跃林，邹福泰，韩佳佳，孙鑫，王一磊。Llm-tikg：利用大型语言模型构建威胁情报知识图谱。可在SSRN 4671345处获得.

[92] 肖恩・巴纳姆。使用结构化威胁信息表达（STIX）标准化网络威胁情报信息。麻省理工学院公司, 11: 1–22, 2012.

[93] 张婷，伊万娜・克莱琳・伊尔桑，费迪安・童，大卫・洛。丘比特：利用ChatGPT实现更准确的重复错误报告检测。arXiv预印本 arXiv:2308.10022, 2023.

[94] 孙承年，卢大卫，邱绍成和姜静。朝着更准确的重复错误报告检索。 在2011年第26届IEEE/ACM国际自动化软件工程会议（ASE 2011），2011年，第253-262页。

[95] 林宇政，穆罕塔西尔・马穆恩，穆赫塔辛・阿拉姆・乔杜里，蔡舒宇，朱明宇，巴纳夫舍・萨贝尔・拉提巴里，凯文・伊曼纽尔・古比，纳吉梅・纳扎里・巴瓦萨德，阿尔贾・卡普托，阿维斯塔・萨桑，侯曼・霍马尤恩，塞塔雷・拉法提拉德，普拉提克・萨塔姆和索黑尔・萨莱希。Hw-v2w-map：硬件漏洞到弱点映射框架，用于带有gpt辅助缓解建议的根本原因分析。arXiv预印本arXiv:2312.13530，2023年。

[96] 梅尔达德・卡赫，丹尼尔・科什・科尔格和帕诺斯・科斯塔科斯。网络哨兵：探索使用GPT-4简化安全任务的对话代理。arXiv预印本 arXiv:2309.16422，2023年。

[97] 金建东，唐博文，马明轩，刘晓，王云飞，赖庆南，杨佳和周长岭。Crimson：通过大型语言模型增强网络安全的战略推理。arXiv预印本arXiv:2403.00878，2024年。

[98] 张颖，宋文佳，季正杰，姚丹峰和孟娜。LLM生成安全测试的效果如何? arXiv预印本 arXiv:2310.00710，2023年。

[99] 胡杰，张倩和尹恒。用生成AI增强灰盒模糊测试。arXiv预印本 arXiv:2306.06782，2023年。

[100] Chunqiu Steven Xia, Matteo Paltenghi, Jia Le Tian, Michael Pradel, and Lingming Zhang. Fuzz4all: Universal fuzzing with large language models.arXiv预印本 arXiv:2308.04748, 2024.

[101] Caroline Lemieux, Jeevana Priya Inala, Shuvendu K. Lahiri, and Siddhartha Sen. Codamosa: 通过预训练的大型语言模型逃离测试生成中的覆盖平台。 在2023年IEEE/ACM第45届国际软件工程大会(ICSE), 页码919-931, 2023年。

[102] Cen Zhang, Mingqiang Bai, Yaowen Zheng, Yeting Li, Xiaofei Xie, Yuekang Li, Wei Ma, Limin Sun, and Yang Liu. 理解基于大型语言模型的模糊驱动程序生成。arXiv预印本 arXiv:2307.12469, 2023年。

[103] 邓寅林，夏春秋，彭浩然，杨晨元，张玲明。大型语言模型是零射击模糊器：通过大型语言模型对深度学习库进行模糊测试。 在第32届ACM SIGSOFT国际软件测试与分析研讨会(ISSTA 2023)论文集，第423-435页，2023年，美国纽约。 计算机协会。

[104] 邓寅林，夏春秋，杨晨元，张世卓，杨书静，张玲明。
大型语言模型是边缘案例模糊器：通过fuzzgpt测试深度学习库。 *arXiv*预印本 *arXiv:2304.02014*，2023年。

[105] 孟瑞杰，马丁・米尔切夫，马塞尔・伯姆，阿比克・罗伊乔德胡里。大型语言模型引导的协议模糊测试。 在第31届年度网络与分布式系统安全研讨会(NDSS)论文集，2024年。

[106] Asmita, Yaroslav Oliinyk, Michael Scott, Ryan Tsang, Chongzhou Fang, and Houman Homayoun. 利用LLM和崩溃重用挖掘嵌入式错误的模糊测试。arXiv预印本 arXiv:2403.03897，2024年。

[107] Anton Cheshkov, Pavel Zadorozhny, and Rodion Levichev. 评估ChatGPT模型用于漏洞检测。arXiv预印本 arXiv:2304.07232，2023年。

[108] Moumita Das Purba, Arpita Ghosh, Benjamin J. Radford, and Bill Chu. 使用大型语言模型进行软件漏洞检测。 在2023年IEEE第34届软件可靠性工程研讨会（ISSREW）的论文集，页码111-119，2023年。

[109] Marwan Omar. 使用语言模型检测软件漏洞。arXiv预印本 arXiv:2302.11773，2023年。

[110] Avishree Khare, Saikat Dutta, Ziyang Li, Alaia Solko-Breslin, Rajeev Alur, and Mayur Naik. 理解大型语言模型在检测安全漏洞中的有效性。arXiv预印本 arXiv:2311.16169,2023.

[111] Rasmus Ingemann Tuffveson Jensen, Vali Tawosi, and Salwa Alamir. 使用LLMs进行软件漏洞和功能评估。arXiv预印本 arXiv:2403.08429, 2024.

[112] Haonan Li, Yu Hao, Yizhuo Zhai, and Zhiyun Qian. 程序分析的搭车人指南：与大型语言模型一起的旅程。arXiv预印本 arXiv:2308.00245, 2023.

[113] Jin Wang, Zishan Huang, Hengli Liu, Nianyi Yang, and Yinhao Xiao. Defecthunter：一种新颖的LLM驱动的基于增强构象的代码漏洞检测机制。arXiv预印本 arXiv:2309.15324, 2023.

[114] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition.arXiv预印本 arXiv:2005.08100, 2020.

[115] Chenyuan Zhang, Hao Liu, Jiutian Zeng, Kejing Yang, Yuhong Li, and Hui Li. Prompt-enhanced software vulnerability detection using chatgpt.arXiv预印本 arXiv:2308.12697, 2023.

[116] Atieh Bakhshandeh, Abdalsamad Keramatfar, Amir Norouzi, and Mohammad Mahdi Chekidehkhoun. Using chatgpt as a static application security testing tool.arXiv预印本 arXiv:2308.14434, 2023.

[117] Noble Saji Mathews, Yelizaveta Brus, Yousra Aafer, Mei Nagappan, and Shane McIntosh. Llbezpeky: 利用大型语言模型进行漏洞检测。arXiv预印本 arXiv:2401.01269, 2024.

[118] Sihao Hu, Tiansheng Huang, Fatih ・Ilhan, Selim Furkan Tekin, and Ling Liu. 大型语言模型驱动的智能合约漏洞检测：新视角。arXiv预印本 arXiv:2310.01152, 2023.

[119] Zhihong Liu, Qing Liao, Wenchao Gu, and Cuiyun Gao. 使用gpt和上下文学习进行软件漏洞检测。在2023年第八届国际数据科学与网络空间会议(DSC), 页码229–236, 2023.

[120] 孙宇强，吴道远，薛悦，刘涵，王海军，徐正子，谢晓飞，刘洋。Gptscan：通过将GPT与程序分析相结合来检测智能合约中的逻辑漏洞。 arXiv预印本arXiv:2308.03314，2023年。

[121] 孙宇强，吴道远，薛悦，刘涵，马伟，张璐烨，石苗磊，刘洋。Llm4vuln：用于解耦和增强LLMs漏洞推理的统一评估框架。arXiv预印本arXiv:2401.16185，2024年。

[122] 毛振宇，李佳龙，李木南，齐健吉。通过LLMs讨论进行多角色共识以进行漏洞检测。arXiv预印本arXiv:2403.14274，2024年。

[123] 陈天宇，李林，朱柳川，李宗洋，梁光泰，李丁，王千祥和谢涛。Vullibgen：通过生成预训练模型识别易受攻击的第三方库。arXiv预印本*arXiv:2308.04662*，2023年。

[124] 刘培宇，刘俊明，傅立荣，陆康杰，夏一凡，张旭宏，陈文智，翁海琴，季守灵和王文海。ChatGPT如何解决漏洞管理问题。arXiv预印本*arXiv:2311.06530*，2023年。

[125] 陈一正，丁洲杰，Lamya Alowain，陈欣云和大卫·瓦格纳。Diversevul：基于深度学习的漏洞检测的新漏洞源代码数据集。 在第26届攻击、入侵和防御研究国际研讨会（RAID’23），第654-668页，美国纽约，2023年。 计算机协会。

[126] Norbert Tihanyi, Tamas Bisztray, Ridhi Jain, Mohamed Amine Ferrag, Lucas C. Cordeiro, and Vasileios Mavroeidis. The formai dataset: Generative ai in software security through the lens of formal verification. In *Proceedings of the 19th International Conference on Predictive Models and Data Analytics in Software Engineering*, PROMISE 2023, page 33–43, New York, NY, USA, 2023. Association for Computing Machinery.

[127] Zeyu Gao, Hao Wang, Yuchen Zhou, Wenyu Zhu, and Chao Zhang. How far have we gone in vulnerability detection using large language models. *arXiv preprint arXiv:2311.12420*, 2023.

[128] Gustavo Sandoval, Hammond Pearce, Teo Nys, Ramesh Karri, Siddharth Garg, and Brendan Dolan-Gavitt. Lost at c: A user study on the security implications of large language model code assistants. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2205–2222, Anaheim, CA, August 2023. USENIX Association.

[129] Florian Tambon, Arghavan Moradi Dakhel, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Giuliano Antoniol. Bugs in large language models generated code: An empirical study. *arXiv preprint arXiv:2403.08937*, 2024.

[130] Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 键盘上犯困了吗？ 评估GitHub Copilot代码贡献的安全性。 在2022年IEEE安全与隐私研讨会（SP），2022年，页码754-768。

[131] 王志龙，张岚，曹晨，刘鹏。大型语言模型（ChatGPT和CodeBERT）在面向安全的代码分析中的有效性。arXiv预印本arXiv:2307.12488, 2023年。

[132] 刘志杰，唐宇天，罗霞普，周玉明，张良峰。再也不需要抬手了？ 评估ChatGPT生成代码的质量。*arXiv*预印本*arXiv:2308.04838*, 2023年。

[133] Mohammed Latif Siddiq 和 Joanna C. S. Santos。生成和祈祷：使用sallms评估llm生成的代码。arXiv预印本 arXiv:2311.00889, 2023.

[134] 刘佳伟、夏春秋、王玉瑶和张玲明。你的代码是由chatgpt生成的吗？ 对代码生成的大型语言模型进行严格评估。神经信息处理系统的进展, 36, 2024.

[135] Saad Ullah、Mingji Han、Saurabh Pujar、Hammond Pearce、Ayse Coskun 和 Gianluca Stringhini。大型语言模型能够识别和推理安全漏洞吗？还没有。arXiv预印本 arXiv:2312.12575, 2023.

[136] Alessio Buscemi。使用chatgpt 3.5在10种编程语言中进行代码生成的比较研究。*arXiv*预印本 *arXiv:2308.04477*, 2023.

[137] Raphaël Khoury, Anderson R. Avila, Jacob Brunelle, and Baba Mamadou Camara. ChatGPT生成的代码有多安全？arXiv预印本 arXiv:2304.09655, 2023.

[138] 何静轩和Martin Vechev。用于代码的大型语言模型：安全加固和对抗测试。 在2023年ACM SIGSAC计算机和通信安全会议论文集, CCS’23. ACM，2023年11月。

[139] Hammond Pearce, Benjamin Tan, Prashanth Krishnamurthy, Farshad Khorrami, Ramesh Karri和Brendan Dolan-Gavitt。小测验！大型语言模型能帮助逆向工程吗？ arXiv预印本*arXiv:2202.01142*, 2022.

[140] 谭瀚卓，罗琦，李静，张宇群。Llm4decompile: 使用大型语言模型反编译二进制代码。arXiv预印本 arXiv:2403.05286，2024年。

[141] 方崇洲，苗宁，Shaurya Srivastav，刘佳琳，张若愚，方瑞杰，Asmita Asmita，Ryan Tsang，Najmeh Nazari，王涵，和Houman Homayoun。用于代码分析的大型语言模型：LLMs真的能胜任吗？arXiv预印本 arXiv:2310.12357，2023年。

[142] 赵建宇，荣宇洋，郭一文，何一峰，和陈浩。通过利用（模糊化）测试用例来理解程序。arXiv预印本 arXiv:2305.13592，2023年。

[143] 大卫·帕拉西奥，亚历杭德罗·韦拉斯科，丹尼尔·罗德里格斯-卡德纳斯，凯文·莫兰和丹尼斯·波什瓦尼克。使用句法结构评估和解释代码的大型语言模型。arXiv预印本 arXiv:2308.03873，2023年。

[144] 闫培，谭顺全，王妙辉和黄继武。使用GPT-4辅助恶意软件动态分析的提示工程。arXiv预印本 arXiv:2312.08317，2023年。

[145] 藤间日玛，熊本多可子和吉田云子。使用ChatGPT分析勒索软件消息并预测勒索软件威胁，2023年。

[146] 王芳。使用大型语言模型减轻勒索软件威胁，2023年。

[147] 努斯拉特·扎汉，菲利普·伯克哈特，米科拉·利森科，费罗斯·阿布哈迪杰和劳瑞·威廉姆斯。转变视角：使用大型语言模型在npm生态系统中检测恶意软件。arXiv预印本 arXiv:2403.12196，2024年。

[148] 刘哲，陈春阳，王俊杰，陈梦卓，吴博宇，车星，王丹丹和王青。将LLM打造成测试专家：通过功能感知决策将人类化互动引入移动GUI测试。
*arXiv*预印本 *arXiv:2310.15780*，2023年。

[149] Baleegh Ahmad，Benjamin Tan，Ramesh Karri和Hammond Pearce。Flag：使用生成AI在代码中查找行异常。arXiv预印本 arXiv:2306.12643，2023年。

[150] Julian Aron Prenner和Romain Robbes。使用OpenAI的Codex进行自动程序修复：评估Quixbugs。*arXiv*预印本 *arXiv:2111.03922*，2021年。

[151] Dominik Sobania，Martin Briesch，Carol Hanna和Justyna Petke。ChatGPT的自动错误修复性能分析。arXiv预印本 arXiv:2301.08653，2023年。

[152] Jan Keller和Jan Nowakowski. Ai-powered patching: the future of automated vulnerability fixes. Technical report, 2024.

[153] Jiaxin Yu，Peng Liang，Yujia Fu，Amjed Tahir，Mojtaba Shahin，Chong Wang和Yangxiao Cai. Security code review by llms: A deep dive into responses.arXiv预印本arXiv:2401.16310, 2024.

[154] Chunqiu Steven Xia，Yuxiang Wei和Lingming Zhang. Practical program repair in the era of large pre-trained language models.arXiv预印本arXiv:2210.14179, 2022.

[155] Hammond Pearce，Benjamin Tan，Baleegh Ahmad，Ramesh Karri和Brendan Dolan-Gavitt. Examining zero-shot vulnerability repair with large language models. 在*2023* IEEE安全与隐私研讨会（*S*P）, 页码*233*9–*23*56, *2023*.

[156] 伍毅，江楠，Pham Viet Hung，Thibaud Lutellier，Jordan Davis，谭琳，Petr Babkin和Sameena Shah。神经网络修复安全漏洞的效果如何。 在第*32*届*ACM* SI*G*S*OF*T国际软件测试与分析研讨会, ISSTA'*23*. *ACM*，2023年7月。

[157] Matthew Jin，Syed Shahriar，Michele Tufano，Xin Shi，Shuai Lu，Neel Sundaresan和Alexey Svyatkovskiy。Inferfix：使用LLMs进行端到端程序修复。arXiv预印本 arXiv:2303.07263, 2023年。

[158] David de Fitero-Dominguez，Eva Garcia-Lopez，Antonio Garcia-Cabot和Jose-Javier Martinez-Herraiz。使用大型语言模型增强自动化代码漏洞修复。arXiv预印本 arXiv:2401.03741, 2024年。

[159] 辛云陈，麦克斯韦林，纳撒尼尔夏利，丹尼周。教大型语言模型自我调试。
*arXiv*预印本 *arXiv:2304.05128*，2023年。

[160] Toufique Ahmed和Premkumar Devanbu。使用llm提示进行更好的修补，通过自一致性。 在2023年第*38*届I*EEE/*A*CM*国际自动化软件工程大会（A*SE*），页面1742-1746，2023年。

[161] 王学智，魏杰森，戴尔舒尔曼斯，乐曲，艾德奇，沙兰纳兰，阿坎夏乔德里和丹尼周。自一致性改善语言模型的思维链推理。 arXiv预印本*arXiv:2203.11171*，2023年。

[162] 雨祥魏，春秋史蒂文夏，张玲明。与完成引擎合并的大型语言模型共同驾驶飞行员：自动程序修复。在第*3*1届*AC*M欧洲联合软件工程大会和软件工程基础研讨会，*ESE*C/FS*E* 202*3*，页面172-184，美国纽约，纽约，202*3*年。 计算机协会。

[163] Nafis Tanveer Islam, Joseph Khoury, Andrew Seong, Mohammad Bahrami Karkevandi, Gonzalo De La Torre Parra, Elias Bou-Harb, and Peyman Najafirad. 利用强化学习和语义奖励修复代码漏洞的LLM动力.arXiv预印本 arXiv:2401.03374, 2024.

[164] Yuxiao Chen, Jingzheng Wu, Xiang Ling, Changjiang Li, Zhiqing Rui, Tianyue Luo, and Yanjun Wu. 当大型语言模型面对存储库级自动程序修复时：他们表现如何？arXiv预印本*arXiv:2403.00448*, 2024.

[165] M. Caner Tol and Berk Sunar. Zeroleak：使用LLM进行可扩展和成本效益的侧信道修补.ar*X*iv预印本 ar*X*iv:*2308*.1*30*6*2*, 2023.

[166] Sudipta Paria, Aritra Dasgupta, and Swarup Bhunia. Divas: 基于LLM的端到端框架用于SOC安全分析和基于策略的保护。ar*X*iv预印本 ar*X*iv:*2308*.*0*6*932*, 2023.

[167] Baleegh Ahmad, Shailja Thakur, Benjamin Tan, Ramesh Karri, and Hammond Pearce. 使用大型语言模型修复硬件安全漏洞。ar*X*i*v*预印本 ar*X*i*v:2302*.*01215*, 2023.

[168] Tan Khang Le, Saba Alimadadi, and Steven Y Ko. 使用大型语言模型修复JavaScript程序中的漏洞修复研究。ar*X*iv电子打印, 页码ar*X*iv–2403, 2024.

[169] Egil Karlsen, Xiao Luo, Nur Zincir-Heywood, and Malcolm Heywood. 为日志分析、安全性和解释性进行大型语言模型基准测试。arXi*v*预印本 arXi*v:23*11.1*45*19, 2023.

[170] 刘金洋，黄俊杰，霍银桐，姜志瀚，顾佳臻，陈壮斌，冯聪，严敏智，以及Michael R. Lyu。基于evt理论和反馈的基于日志的异常检测。 ar*X*iv预印本*arXiv:2306.05032*，2023年。

[171] 齐佳兴，黄少瀚，栾忠志，冯卡罗，杨海龙，以及钱德培。Loggpt: 探索基于聊天GPT的基于日志的异常检测。ar*X*i*v*预印本ar*X*i*v:2309*.*0*11*89*，*2023*年。

[172] 韩晓，袁舒涵，以及Mohamed Trabelsi。Loggpt: 通过GPT进行日志异常检测。arXiv预印本arXiv:2*309*.1*448*2，2*02*3年。

[173] Yilun Liu, Shimin Tao, Weibin Meng, Jingyu Wang, Wenbing Ma, Yanqing Zhao, Yuhang Chen, Hao Yang, Yanfei Jiang, and Xun Chen. 使用大型语言模型和提示策略进行可解释的在线日志分析。*arXiv*预印本 *arXiv:2308.07610*, 2024.

[174] Wei Zhang, Hongcheng Guo, Anjie Le, Jian Yang, Jiaheng Liu, Zhoujun Li, Tieqiao Zheng, Shi Xu, Runqiang Zang, Liangfan Zheng, and Bo Zhang. Lemur: 使用熵采样和思维链合并进行日志解析。*arXiv*预印本 *arXiv:2402.18205*, 2024.

[175] Tamás Vörös, Sean Paul Bergeron, and Konstantin Berlin. 通过大型语言模型的知识蒸馏进行网络内容过滤。ar*X*i*v*预印本 ar*X*i*v:230*5.*05027*, 2023.

[176] Michael Guastalla, Yiyi Li, Arvin Hekmati, and Bhaskar Krishnamachari. 大型语言模型在DDoS攻击检测中的应用。

[177] Suhaima Jamal and Hayden Wimmer. 用于检测钓鱼、垃圾邮件和正常邮件的改进的基于Transformer的模型：一种大型语言模型方法。ar*X*iv预印本 ar*X*iv:*23*11.*04*913*, 2023*年。

[178] Yuwei Wu, Shijing Si, Yugui Zhang, Jiawen Gu, and Jedrek Wosik. 评估ChatGPT在垃圾邮件检测中的性能。ar*X*iv预印本 ar*X*iv*:2402*.1*553*7, *2024*年。

[179] Daniel Nahmias, Gal Engelberg, Dan Klein, and Asaf Shabtai. 用于钓鱼邮件检测的提示性上下文向量。ar*X*iv预印本 ar*X*iv*:2402*.*0830*9, *2024*年。

[180] Fredrik Heiding, Bruce Schneier, Arun Vishwanath, Jeremy Bernstein, and Peter S. Park. 设计和检测网络钓鱼：大型语言模型与较小的人类模型。ar*X*iv预印本 ar*X*iv:*23*08.1*2287*，2*023*年。

[181] Noah Ziems, Gang Liu, John Flanagan, and Meng Jiang. 解释自然语言中树模型决策用于网络入侵检测。ar*X*i*v*预印本 ar*X*i*v:231*0.1*9658*，*2023*年。

[182] Mohamed Amine Ferrag, Mthandazo Ndhlovu, Norbert Tihanyi, Lucas C. Cordeiro, Merouane Debbah, and Thierry Lestable. 用大型语言模型彻底改变网络威胁检测：面向物联网/工业物联网设备的保护隐私的基于bert的轻量级模型。ar*X*iv预印本 ar*X*iv:2*306*.1*4263*，2*02*3年。

[183] Tarek Ali 和 Panos Kostakos。Huntgpt：将基于机器学习的异常检测和可解释人工智能与大型语言模型（LLMs）集成。ar*X*i*v*预印本 ar*X*i*v*:*2402*.18*205*，*202*3年。

[184] Mark Scanlon，Frank Breitinger，Christopher Hargreaves，Jan-Niclas Hilgert 和 John Sheppard。Chatgpt 用于数字取证调查：好的、坏的和未知的。法医科学国际：数字调查，46:301609，2023年。

[185] Clark Barrett，Brad Boyd，Elie Bursztein，Nicholas Carlini，Brad Chen，Jihye Choi，Amrita Roy Chowdhury，Mihai Christodorescu，Anupam Datta，Soheil Feizi 等。识别和减轻生成式人工智能的安全风险。*Foundations and Trends® in Privacy and Security*，6(1):1–52，2023年。

[186] Pawankumar Sharma 和 Bibhu Dash。大数据分析和 chatgpt 对网络安全的影响。 在2023年第四届计算与通信系统国际会议（I3CS），页码1-6，2023年。

[187] Maanak Gupta，Charankumar Akiri，Kshitiz Aryal，Eli Parker 和 Lopamudra Praharaj。从 chatgpt 到 threatgpt：生成式人工智能在网络安全和隐私中的影响。IEEE Access，11：80218-80245，2023年。

[188] Stephen Moskal，Sam Laney，Erik Hemberg 和 Una-May O’ Reilly。LLMs 打败了脚本小子：受大型语言模型支持的代理如何改变网络威胁测试的格局。 arXiv 预印本arXiv:2310.06936，2023年。

[189] Zilong Lin，Jian Cui，Xiaojing Liao 和 XiaoFeng Wang。Malla：揭秘现实世界中集成恶意服务的大型语言模型。arXiv 预印本 arXiv:2401.03315，2024年。

[190] Andreas Happe, Aaron Kaplan, and Jürgen Cito. 评估LLMs在特权升级场景中的应用。 arXiv 预印本 arXiv:2310.11409, 2023.

[191] Wesley Tann, 刘远成, Jun Heng Sim, Choon Meng Seah, 和 Ee-Chien Chang. 使用大型语言模型进行网络安全夺旗挑战和认证问题。arXiv 预印本 arXiv:2308.10443,2023.

[192] Nils Begou, Jérémy Vinoy, Andrzej Duda, 和 Maciej Korczy´nski. 探索人工智能的黑暗面：使用ChatGPT进行高级网络钓鱼攻击设计和部署。 在2023年IEEE通信和网络安全会议, 页码1–6, 2023.

[193] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, 和 Shirin Nilizadeh. 从聊天机器人到网络钓鱼机器人？ – 防止使用ChatGPT、Google Bard和Claude创建的网络钓鱼诈骗。arXiv 预印本 arXiv:2310.19181,2024.

[194] P. V. Sai Charan, Hrushikesh Chunduri, P. Mohan Anand, and Sandeep K Shukla. 从文本到mitre技术：探索大型语言模型用于生成网络攻击载荷的恶意用途。 arXiv预印本arXiv:2305.15336，2023年。

[195] 邓格雷，刘毅，维克多・马约拉尔-维尔切斯，刘鹏，李跃康，徐远，张天伟，刘洋，马丁・平兹格，斯特凡・拉斯。Pentestgpt：一种llm增强的自动渗透测试工具。arXiv预印本arXiv:2308.06782，2023年。

[196] 安德烈亚斯・哈佩和尤尔根・西托。被人工智能pwn’d：使用大型语言模型进行渗透测试。 在第31届ACM欧洲联合软件工程大会和软件工程基础研讨会（ESEC/FSE 2023）论文集中，第2082-2086页，2023年，美国纽约。 计算机协会。

[197] 许家岑，杰克・斯托克斯，杰夫・麦克唐纳，白学松，大卫・马歇尔，王思悦，阿迪斯・斯瓦米纳坦，和周立。Autoattacker：一个大型语言模型引导系统，用于实现自动网络攻击。arXiv预印本 arXiv:2403.01038，2024年。

[198] 米卡・贝克里奇，劳拉・普莱恩，和塞尔吉奥・科罗纳多。Ratgpt：将在线LLMs转化为恶意软件攻击的代理。arXiv预印本 arXiv:2308.09183，2023年。

[199] 阿明・萨拉比，尹同鑫，和刘明燕。基于LLM的互联网设备指纹识别框架。 在2023年ACM互联网测量会议论文集，IMC’23，第478–484页，美国纽约，2023年。 计算机协会。

[200] Kai-Cheng Yang 和 Filippo Menczer。Anatomy of an ai-powered malicious social botnet。 arXiv 预印本 arXiv:2307.16336, 2023。

[201] Xunzhu Tang, Zhenghan Chen, Kisub Kim, Haoye Tian, Saad Ezzini, 和 Jacques Klein。Just-in-time security patch detection – llm at the rescue for data augmentation。arXiv 预印本 arXiv:2312.01241, 2023。

[202] Dipayan Saha, Shams Tarek, Katayoon Yahyaei, Sujan Kumar Saha, Jingbo Zhou, Mark Tehranipoor, 和 Farimah Farahmandi。Llm for soc security: A paradigm shift。arXiv 预印本 arXiv:2310.06046, 2023。

[203] Puzhuo Liu, Chengnian Sun, Yaowen Zheng, Xuan Feng, Chuan Qin, Yuncheng Wang, Zhi Li, 和 Limin Sun。Harnessing the power of llm to support binary taint analysis。arXiv 预印本 arXiv:2310.08275, 2023。

[204] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, 和 Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations.arXiv 预印本 arXiv:2312.06674, 2023.

[205] Muris Sladi´c, Veronica Valeros, Carlos Catania, 和 Sebastian Garcia. Llm in the shell: Generative honeypots. arXiv 预印本 arXiv:2309.00155, 2024.

[206] Sam Hays 和 Dr. Jules White. Employing llms for incident response planning and review. arXiv 预印本 arXiv:2403.01271, 2024.

[207] Sathiya Kumaran Mani, Yajie Zhou, Kevin Hsieh, Santiago Segarra, Trevor Eberl, Eliran Azulai, Ido Frizler, Ranveer Chandra, and Srikanth Kandula. 利用大型语言模型生成的代码增强网络管理。在第22届ACM热点网络研讨会论文集中, HotNets '23, 页码196–204, 美国纽约, 2023年。 计算机协会。

[208] 冯思东和陈春阳。提示就是你所需要的：利用大型语言模型自动重放安卓漏洞。arXiv预印本 arXiv:2306.01987, 2023年。

[209] Samia Kabir, David N. Udo-Imeh, Bonan Kou, 和 Tianyi Zhang。堆栈溢出是否已过时？ 对ChatGPT回答Stack Overflow问题特征的实证研究。arXiv预印本 arXiv:2308.02312, 2024年。

[210] 赵宣东，杨显军，庞天宇，杜超，李磊，王宇翔和William Yang Wang。
大型语言模型上的弱到强越狱。arXiv预印本 arXiv:2401.17256，2024年。

[211] Surender Suresh Kumar，ML Cummings和Alexander Stimpson。加强llm信任边界：提示注入攻击调查。

[212] Aysan Esmradi，Daniel Wankit Yip和Chun Fai Chan。大型语言模型中攻击技术、实施和缓解策略的综合调查。arXiv预印本 arXiv:2312.10982，2023年。

[213] 吴方舟，张宁，Somesh Jha，Patrick McDaniel和Chaowei Xiao。llm安全的新时代：探索现实世界llm系统中的安全问题。arXiv预印本 arXiv:2402.18649，2024年。

[214] Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. 对LLMs的越狱攻击进行全面评估。arXiv预印本 arXiv:2402.05668, 2024.

[215] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. LLM越狱攻击与防御技术的综合研究。arXiv预印本 arXiv:2402.13457, 2024.

[216] Jiawen Shi, Yixin Liu, Pan Zhou, and Lichao Sun. Badgpt: 通过后门攻击探索ChatGPT的安全漏洞以指导GPT。arXiv预印本 arXiv:2304.12298, 2023.

[217] Shuai Zhao, Meihuizi Jia, Luu Anh Tuan, Fengjun Pan, and Jinming Wen. 大型语言模型的通用漏洞：用于上下文学习的后门攻击。arXiv预印本 arXiv:2401.05949, 2024.

[218] 姚宏伟，楼健，秦展。Poisonprompt：基于提示的大型语言模型的后门攻击。
*arXiv*预印本 *arXiv:2310.12439*，2023年。

[219] Rodrigo Pedro，Daniel Castro，Paulo Carreira，Nuno Santos。从提示注入到SQL注入攻击：您的LLM集成Web应用程序有多安全？arXiv预印本 arXiv:2308.01990，2023年。

[220] 姜树宇，陈兴树，唐锐。Prompt packer：通过隐藏攻击的组合指令欺骗LLMs。arXiv预印本 arXiv:2310.10077，2023年。

[221] 刘宇培，贾宇琦，耿润鹏，贾金元，宫振强。LLM集成应用程序中的提示注入攻击和防御。arXiv预印本 arXiv:2310.12815，2023年。

[222] Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, 和 Hongxia Jin. 通过虚拟提示注入对调整后的大型语言模型进行后门操作。arXiv预印本 *arXiv:2307.16888*, 2023.

[223] Julien Piet, Maha Alrashed, Chawin Sitawarin, Sizhe Chen, Zeming Wei, Elizabeth Sun, Basel Alomair, 和 David Wagner. Jatmo: 通过任务特定的微调进行提示注入防御。arXiv预印本 arXiv:2312.17673,2024.

[224] George Kour, Marcel Zalmanovici, Naama Zwerdling, Esther Goldbraich, Ora Nova Fandina, Ateret Anaby- Tavor, Orna Raz, 和 Eitan Farchi. 揭示大型语言模型的安全漏洞。arXiv预印本*arXiv:2311.04124*, 2023.

[225] 辛悦，陈泽远，迈克尔·巴克斯，沈云和张洋。"现在做任何事情": 对大型语言模型中的野外越狱提示进行表征和评估。arXiv预印本 arXiv:2308.03825，2023年。

[226] 安迪·邹，王子凡，尼古拉斯·卡林尼，米拉德·纳斯尔，J. Zico Kolter和马特·弗雷德里克森。通用和可转移的对齐语言模型的对抗攻击。arXiv预印本 arXiv:2307.15043，2023年。

[227] Raz Lapid，Ron Langberg和Moshe Sipper。开门! 大型语言模型的通用黑盒越狱。arXiv预印本 arXiv:2309.01446，2023年。

[228] 丁鹏，匡俊，马丹，曹学志，冼云森，陈佳俊和黄树坚。披着羊皮的狼: 广义嵌套越狱提示可以轻松愚弄大型语言模型。arXiv预印本*arXiv:2311.08268*，2024年。

[229] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Masterkey: 大型语言模型聊天机器人的自动越狱。 在2024年网络和分布式系统安全研讨会（NDSS 2024）的论文集中，互联网协会，2024年。

[230] Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. Autodan: 可解释的基于梯度的大型语言模型对抗攻击。arXiv预印本arXiv:2310.15140, 2023.

[231] Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. Gptfuzzer: 使用自动生成的越狱提示对大型语言模型进行红队行动。arXiv预印本 arXiv:2309.10253, 2023.

[232] 姚东宇，张建树，伊恩・哈里斯和马塞尔・卡尔松。Fuzzllm：一种新颖且通用的模糊测试框架，用于主动发现大型语言模型中的越狱漏洞。 arXiv预印本arXiv:2309.05274，2023年。

[233] 王振华，谢伟，陈凯，王宝生，桂志文和王恩泽。自欺欺人：逆向渗透大型语言模型的语义防火墙。arXiv预印本arXiv:2308.11521，2023年。

[234] 邱华川，张帅，李安琪，何洪亮和兰振中。潜在越狱：用于评估大型语言模型文本安全性和输出稳健性的基准。arXiv预印本arXiv:2307.08487，2023年。

[235] 郝然李，郭大地，范伟，徐明石，黄杰，孟凡普，宋阳秋。ChatGPT上的多步越狱隐私攻击。arXiv预印本 arXiv:2304.05197，2023年。

[236] Rajesh Pasupuleti，Ravi Vadapalli和Christopher Mader。与生成AI和ChatGPT相关的网络安全问题和挑战。 在2023年第十届社交网络分析、管理和安全国际会议（SNAMS），2023年，第1-5页。

[237] Evan Hubinger，Carson Denison，Jesse Mu，Mike Lambert，Meg Tong，Monte MacDiarmid，Tamera Lanham，Daniel M. Ziegler，Tim Maxwell，Newton Cheng，Adam Jermyn，Amanda Askell，Ansh Radhakrishnan，Cem Anil，David Duvenaud，Deep Ganguli，Fazl Barez，Jack Clark，Kamal Ndousse，Kshitij Sachan，Michael Sellitto，Mrinank Sharma，Nova DasSarma，Roger Grosse，Shauna Kravec，Yuntao Bai，Zachary Witten，Marina Favaro，Jan Brauner，Holden Karnofsky，Paul Christiano，Samuel R. Bowman，Logan Graham，Jared Kaplan，Sören Mindermann，Ryan Greenblatt，Buck Shlegeris，Nicholas Schiefer和Ethan Perez。沉睡特工：训练经过安全训练的具有持久性的欺骗LLMs。arXiv预印本 arXiv:2401.05566，2024年。

[238] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 影子对齐：绕过安全对齐语言模型的简易方法。arXiv预印本 arXiv:2310.02949, 2023.

[239] Fengqing Jiang, Zhangchen Xu, Luyao Niu, Boxin Wang, Jinyuan Jia, Bo Li, and Radha Poovendran. 识别和减轻LLM集成应用程序中的漏洞。arXiv预印本 arXiv:2311.16153, 2023.

[240] June Sallou, Thomas Durieux, and Annibale Panichella. 打破沉默：在软件工程中使用LLMs的威胁。arXiv预印本 arXiv:2312.08055, 2023.

[241] Neda Azizi和Omid Haass。网络安全问题和挑战。 在网络安全问题和挑战研究手册, 页码21-48。IGI Global，2023年。

[242] Jabu Mtsweni, Noluxolo Gcaza, and Mphahlele Thaba. 一个统一的复杂环境网络安全框架。 在南非计算机科学家和信息技术专家年会论文集中, 2018年，第1-9页。

[243] Tanay Varshney. LLM代理简介，2023年。

[244] 崔洪伟，杜宇洋，杨群，邵玉林和刘松昌。LLMind：用LLM协调AI和物联网进行复杂任务执行。arXiv预印本 arXiv:2312.09007, 2024年。

[245] Maria Rigaki, Ondˇrej Lukáš, Carlos A. Catania, and Sebastian Garcia. 出笼：随机鹦鹉在网络安全环境中获胜。arXiv预印本 arXiv:2308.12086, 2023年。

[246] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. LLM代理可以自主黑客网站。arXiv预印本 arXiv:2402.06664, 2024.

[247] Kaikai An, Fangkai Yang, Liqun Li, Zhixing Ren, Hao Huang, Lu Wang, Pu Zhao, Yu Kang, Hua Ding, Qingwei Lin, Saravan Rajmohan, and Qi Zhang. Nissist: 基于故障排除指南的事件缓解副驾驶员。arXiv预印本 arXiv:2402.17531, 2024.

[248] Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Guoqing Du, Shiwei Shi, Hangyu Mao, Ziyue Li, Xingyu Zeng, and Rui Zhao. Tptu: 基于大型语言模型的人工智能代理进行任务规划和工具使用。arXiv预印本 arXiv:2308.03427, 2023.

[249] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. 基于大型语言模型的代理的崛起和潜力：一项调查。arXiv预印本 arXiv:2309.07864, 2023.

[250] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm：促进大型语言模型掌握16000多个真实世界的API。arXiv预印本*arXiv:2307.16789*, 2023.

[251] 刘玉龙，袁云龙，王春伟，韩建华，马勇强，张力，郑南宁，徐航。从摘要到行动：利用开放世界API增强大型语言模型的复杂任务。 *arXiv*预印本 arXiv:2402.18157，2024年。

[252] 杨科，刘佳腾，吴约翰，杨超奇，冯一然，李莎，黄子轩，曹旭，王兴耀，王一权，季恒，翟成祥。如果LLM是巫师，那么代码就是魔杖：关于代码如何赋予大型语言模型作为智能代理的调查。arXiv预印本 arXiv:2401.00812，2024年。

[253] 乔波，李立群，张旭，何世林，康宇，张朝云，杨方凯，董航，张觉，王璐，马明华，赵璞，秦思，秦晓婷，杜超，徐勇，林庆伟，拉杰莫汉，张东梅。Taskweaver：一个以代码为先的代理框架。arXiv预印本 arXiv:2311.17541，2023年。

[254] 黄宇东，杜洪洋，张新元，Dusit Niyato，康佳文，熊泽辉，王硕，黄涛。大型语言模型在网络中的应用，启用技术和挑战。arXiv预印本 arXiv:2311.17474，2023年。

[255] 袁同新，何志伟，董灵忠，王一鸣，赵瑞杰，夏天，徐丽珍，周炳林，李方琦，张卓升，王锐，刘功深。R-judge：对llm代理的安全风险意识进行基准测试。arXiv预印本 arXiv:2401.10019，2024年。

[256] 吴方舟，吴树彤，曹宇龙，肖超伟。Wipi：llm驱动的网络代理的新网络威胁。*arXiv*预印本 *arXiv:2402.16965*, 2024.

[257] 詹秋思，梁志祥，应子凡，和丹尼尔·康。Injecagent：在工具集成的大型语言模型代理中进行间接提示注入的基准测试。arXiv预印本 arXiv:2403.02691, 2024.