STACCO:差异分析旁道踪迹以检测安全飞地中的SSL/TLS漏洞

**摘要：**

英特尔软件防护扩展（SGX）为软件应用程序提供了一个被屏蔽的执行环境，称为飞地，以保护其机密性和完整性免受恶意操作系统的侵害。随着具有此扩展功能的处理器在市场上销售，许多新的软件应用程序将开发出来，充实到支持SGX的软件生态系统。这些应用程序的一个重要原语是飞地与远程信任方之间的安全通信通道。作为保护传输层的网络通信的事实上的标准的SSL/TLS协议，一般被认为是这种目的的自然选择。然而，在本文中，我们表明SGX和SSL之间的结合可能不是一帆风顺的。

特别是，我们考虑在安全飞地中针对SSL/TLS实施的一类旁道攻击，我们称之为控制流推理攻击。在这些攻击中，恶意操作系统内核可能会执行强大的内核攻击，以在页面级别，缓存行级别或分支级别收集安全区程序的执行痕迹，同时将自身定位在两个通信方。我们工作的核心是一个名为STACCO的差异分析框架，用于动态分析SSL/TLS实现并检测漏洞-可识别的执行痕迹-可用作解密预测。令人惊讶的是，尽管许多加密库采用了流行的恒定时间编程模式，但我们在所检查的所有SSL/TLS库的最新版本中发现了可利用的漏洞。

为了验证检测到的漏洞，我们开发了一个内核攻击者来展示Bleichenbacher针对在SGX飞地中运行的最新OpenSSL库（在Graphene的帮助下）的攻击，并只用57286条查询完全打破了由4096位RSA公钥加密的PreMasterSecret。我们还对Graphene-SGX中运行的最新GnuTLS和直接在飞地内运行的mbedTLS（即mbedTLS-SGX）的开源SGX实现进行了CBC填充oracle攻击，并表明它分别只需要48,388和25,717条查询来打破一块AES密文。经验性评估表明，这些内核攻击可以在一两个小时内完成。

**1 INTRODUCTION**

软件应用程序的安全性很大程度上取决于底层系统软件的安全性。在传统的计算环境中，如果操作系统遭到破坏，它支持的应用程序的安全性也会受到影响。因此，软件应用程序的可信计算基础（TCB）不仅包括软件本身，还包括底层系统软件和硬件。

为了减少某些包含敏感代码和数据的应用程序的TCB，学术研究人员提出了许多软件系统来支持屏蔽执行-即执行一段代码，其机密性和完整性受到不可信系统软件的保护（例如[26，28，30，33，37，44，45，49，56，68，74，77]）。这些系统中的大多数采用基于管理程序的方法来保护受害应用程序的内存免受恶意操作系统的攻击。虽然很有前途，但这些学术原型还没有无法在现实世界应用。直到英特尔软件防护扩展（SGX）[2]（最新的英特尔处理器中提供的硬件扩展）的出现，屏蔽执行的概念才变得对现实世界的应用更为实用。SGX通过将其存储空间的区域（即，飞地）与其他软件组件（包括最高权限的系统软件）隔离，来强制执行用户空间程序的机密性和完整性-不管存储器读或写都可以通过任何外部软件在飞地内执行，无论的特权级别。因此，SGX大大减少了屏蔽执行的TCB，从而实现了广泛的应用[20，36，54，59，65，76]。

在典型的应用场景[20，36，76]中，屏蔽执行不能完全独立工作；它使用安全信道（例如SSL/TLS协议）与远程信任方进行通信。安全套接字层（SSL）及其后续传输层安全性（TLS）是传输层安全协议，它使用一组加密基元提供安全通信通道。作为其设计目标的一部分，SSL/TLS协议预期能够防止能够在两个通信方之间窃听，拦截，重放，修改和注入网络数据包的中间人攻击者。因此，英特尔SGX[20，36，76]的应用通常将SGX飞地内的SSL/TLS模块视为基本安全原语，以建立端到端通信安全。

不幸的是，多年来已经报告了很多针对SSL/TLS协议的攻击。这些攻击的一个重要类别是Oracle攻击[31]。在oracle攻击中，攻击者交互式地自适应地查询易受攻击的SSL/TLS实现，并将该响应（或某些旁路通道信息，例如响应的延迟）作为oracle破解加密。著名的Oracle攻击示例包括幸运十三攻击[12]，Bleichenbacher攻击[22]，DROWN攻击[18]，POODLE攻击[52]等。这些攻击的先前演示表明，它们启用了网络攻击者解密SSL记录协议的任意消息或解密SSL握手协议的PreMasterSecret。我们将在第2部分详细描述这些攻击。由于SSL/TLS协议的广泛采用（例如HTTPS，安全电子邮件交换），任何这些攻击都是破坏性的，并且容易引发安全新闻的头条新闻（例如[41]）。相应地，SSL/TLS协议及其实现在这些攻击的公开后经常更新。常用的解决方案是隐藏oracles。例如，如果oracle是指示填充错误的SSL警报消息，则可以统一错误消息以隐藏错误的真正原因[35,58]（以使对手无法区分填充错误和MAC错误，请参阅第2部分）。到目前为止，几乎所有广泛使用的SSL/TLS实现都对Oracle攻击具有适应能力，因为Oracle已成功地对网络攻击者隐藏了[4,10,35,58]。

但是，在SGX飞地中采用SSL/TLS带来了新的安全挑战。尽管SGX通过内存隔离和加密，为安全飞地内的代码和数据提供机密性保护，但它已被证明容易受到旁道攻击[43,63,73]。旁道攻击是针对系统或应用的机密性的一种安全攻击，通过从可观测的旁道事件的测量结果推断出来。在过去的二十年中，这些攻击在多种环境中得到了研究，尤其是在台式机，云服务器和移动设备中，CPU微架构[78,79]，软件数据结构[40,57]或其他系统资源在相互不信任的软件组件之间共享。对SGX进行旁道攻击的不同之处在于，这些攻击可以通过特权系统软件来执行，这使得许多新的攻击媒介成为可能。例如，徐等人[73]证明，通过操纵安全区域内存页面的页表条目，具有系统特权的攻击者可以在执行区域程序期间强制执行页面错误，从而收集页面粒度的内存访问痕迹。最近，Lee等人[43]通过利用共享的分支预测单元（BPU），可以在每个分支指令中精确地追踪飞地程序的控制流。

本文的关键洞察在于，尽管SSL/TLS旨在抵御中间人攻击，但在SGX飞地中的实施必须解决更强大的内核对手问题，该对手不仅能够定位他本人在两个通信方中间，而且能控制底层操作系统内核并操纵系统资源以收集来自各个旁道的飞地程序的执行踪迹。特别是，我们证明强大的内核攻击者可以从最先进的SSL/TLS实现中创建新的解密oracle，并恢复Bleichenbacher攻击和CBC填充甲骨文攻击SGX飞地。

**STACCO：**我们工作的核心是用于寻找Chosen-Ciphertext Oracles（STACCO）的旁道踪迹分析器，该分析器是一个软件框架，用于对SSL/TLS实施进行差异分析，以检测可用于敏感的控制流漏洞为CBC填充Oracle攻击和Bleichenbacher攻击创建解密oracle。特别是，为了能够对各种现成的SSL/TLS库进行自动化的大规模分析，我们在动态仪器引擎（如Pin[46]）和开源SSL/TLS数据包生成工具之上构建了STACCO（即TLS攻击者[66]），以便我们能够以自动化的方式对多个库执行标准测试。为了理解这些漏洞的可利用性，我们还建模了三种类型的控制流推理攻击，包括页面级攻击[63,73]，缓存线级攻击[23,60]和分支级攻击[43]，以及授权STACCO分析每个级别的漏洞。我们的分析结果表明，我们已经检查过的所有流行的开源SSL/TLS库都容易受到这两种类型的oracle攻击的影响，引发了SGX飞地内安全开发和SSL/TLS协议部署的问题。

为了验证由STACCO识别的漏洞，我们演示了几种针对最新版本流行密码库的此类内核攻击：特别是，我们针对在SGX飞地中运行的最新OpenSSL库[9]实施了Bleichenbarcher攻击（在Graphene-SGX[70]的帮助下，一个库操作系统支持未经修改的应用程序在SGX飞地内部运行），并只用57,286条查询完全破解了由4096位RSA公钥加密的PreMasterSecret。我们还对Graphene-SGX中运行的最新GnuTLS[3]和直接在飞地内运行的开源SGX实现的mbedTLS[8]进行了CBC填充oracle攻击，并表明它分别使用这些库从TLS连接中分解一段AES密文的过程，只需要48,388和25,717个查询。经验性评估表明，这些内核攻击可以在一两个小时内完成。这些演示攻击不仅证明了STACCO可以有效识别SSL/TLS实施中可利用的敏感控制流漏洞，而且还表明这些以内核方式进行的oracle攻击对于实际安全入侵是有效的。

**负责任的披露：**我们已向英特尔，OpenSSL，GnuTLS，mbedTLS报告了这些漏洞并演示了Oracle攻击。这项工作的贡献包括：

* 针对SGX飞地中SSL/TLS实施的关键旁道威胁进行的首次研究导致SSL/TLS保护的安全通信的完全妥协。
* 设计和实施STACCO，这是一种用于检测SSL/TLS实施中的敏感控制流漏洞的差异分析框架，它还需要：
* 针对SGX飞地（例如，页面级攻击，缓存线级攻击和分支级攻击）的控制流推理攻击的系统特性，使STACCO能够利用抽象的攻击者模型分析漏洞。
* 使用STACCO对最新版本的流行SSL/TLS库进行了测量研究，结果表明，所有这些库（包括OpenSSL，GnuTLS，mbedTLS，WolfSSL和LibreSSL）都容易受到控制流推理攻击，并可在Oracle攻击中利用。
* 针对在Graphene-SGX内部运行的最新版本的OpenSSL和GnuTLS以及开源的SGX实现的mbedTLS的甲骨文攻击的经验性内核演示，显示此类攻击在真正的SGX硬件上非常高效。

**概要（路线图）：**本文的其余部分概述如下。第2节介绍相关的背景概念。第3节系统地描述控制流推理攻击。第4节介绍了一种用于检测SSL/TLS实现中的敏感控制流漏洞的差异分析框架。我们在第5节中演示了针对某些易受攻击的SSL/TLS实现的Oracle攻击来验证这些检测到的漏洞，然后在第6节中讨论对策。在第7节中，我们简要总结了该领域的相关工作。第8节总结了我们的论文。

**2 BACKGROUND**

**2.1 Intel Software Guard Extension**

**2.2 SSL/TLS**

**2.3 Bleichenbacher Attacks against SSL/TLS**

**2.4 CBC Padding Oracle Attacks**