Graphene-SGX

zhuangh7

2019 年 12 月 13 日

1 Introduction

Intel SGX 技术提供了一系列的硬件特性，帮助保护应用免受操作系统、hypervisor、BIOS 和其他软件的伤害。应用程序可以整个或者部分的放入到 enclave 中运行，而 enclave 会有如下功能

- 对 enclave 内的虚拟地址空间的机密性和完整性进行保护。
- 限制控制流进入 enclave 的入口。
- 启动时检查内存的内容。
- 远程证明。

正是因为提供了这些功能，因此用户就可以放心的将他们的敏感应用或者敏感数据放在云服务提供商的 enclave 中去运行，保证自己敏感应用的安全性，大大增加了云服务能够服务的对象。

SGX 的适配较为困难 为了保证 SGX 的安全性，在 SGX 中运行的应用将有一些限制。比如应用不能在 enclave 中调用 syscall，而是通过一层 shielding code 去访问 syscall，通过 sheld 层，syscall 的结果在返回给应用之前会被验证，保证结果的安全可靠，防止 host 的操作系统攻击 enclave 中的应用。因此，普遍认为应用程序适配 SGX 需要付出大量的努力。前人的工作 Haven [1] 展示了可以利用一个 library OS 把没有修改过的应用放到 SGX 中去。但是这需要付出性能上的 overhead 和 TCB 的扩大的代价。不过，Graphene-SGX 验证了实际上利用 library OS 带来的 overhead 并没有前人所称的那么大，利用 Graphene-SGX 可以快速的把现有的应用放到 enclave 中去运行，并且没有很大的开销。

Graphene-SGX 并不是最优的 Graphene-SGX 这一工作的目的并不是为了追求更好的 SGX 技术利用性能，而是为了把已有工作更快的部署到 enclave 中去。因此，虽然有很多方法可以提升 enclave 代码中的性能，降低代码的 TCB，但是 enclave 都暂未采用，因为这些优化会令应用的适配更加复杂。

2 威胁模型

Graphene-SGX 的威胁模型与典型的 SGX 应用的威胁模型类似。下列的组件是不被信任的：

1) 除 CPU 之外的硬件。

2) 操作系统、hypervisor 和其他的系统软件。

3) 在 enclave 之外跟其他 enclave 之内运行的应用程序。

4) 同一应用程序的 enclave 之外的部分。

Graphene-SGX 只信任 CPU，还有 enclave 内部的代码（shielding module、bootloader）。还必须信任 Intel 的 SGX SDK 中的 aesmd 工具，该工具验证 enclave 签名中的属性并批准 enclave 的创建。这是现在利用 SGX 技术的所有工作都必须信任的。除此之外，虽然 Graphene-SGX 使用且仅使用了 Intel SGX SDK 中的驱动程序，但是并不信任他。

3 保证的安全性质和能够防御的一些具体攻击

3.1 保证的安全性质

继承自 SGX 的性质 Graphene-SGX 是对 SGX 技术的应用，因此，Intel SGX 技术提供的保护也被完全的继承。1) 对 enclave 内的虚拟地址空间的机密性和完整性进行保护。2) 限制控制流进入 enclave 的入口。3) 启动时检查内存的内容。4) 连接 Intel 服务器进行远程证明

Graphene-SGX 提供的特性 除此之外，Graphene-SGX 还提供了很多其他安全性质。例如：

- 通过本地验证证明两个本地 enclave 是安全的。
- 在两个安全的本地 enclave 之间搭建安全通信。
- 提供安全的 fork 机制创建一个新的 enclave。

3.2 能够防御的具体攻击

1) **来自 host 上其他进程的攻击** memory corruption,ROP attack

2) **自 host OS 的攻击** rootkits

3) **来自硬件的攻击** cold-boot attacks

4 性能、可扩展性（scalability)、灵活性等分析

4.1 灵活性和可扩展性

由于 Graphene-SGX 可以直接在 enclave 上运行没有经过修改的 Linux 程序，因此，通过 Graphene-SGX 可以快速的帮助现有应用利用到 SGX 的 enclave 这一特性，提高已有应用的安全性，而同时不需要付出太大的开发成本。

在可扩展性方面，Graphene-SGX 还提供了 enclave 的 fork 操作，用户可以简单的通过一个 enclave 环境，fork 出一个新的子 enclave 环境。同时 Graphene-SGX 还提供了两个 enclave 之间的加密通信，互相验证的功能。因此，多进程的应用程序可以有效的保留多进程抽象的同时利用 SGX 技术带来的安全保证。

4.2 性能分析

图 1展示了 Graphene-SGX 的性能分析。可以从途中看出，对轻量级的网站服务应用 **Lighttpd** 来说, Graphene-SGX 对性能的影响不大。对 **Apache** 来说，由于 Apache 服务器有多个 worker，每个 worker 在一个 enclave 环境中，因此 worker 之间的通信，请求的传输，都需要用到 enclave 之间的加密通信，会消耗很大的性能。因此，如果不用 SGX 的话，单独的 Graphenen 技术并不会带来多少的 overhead，但是如果要保证安全性，对性

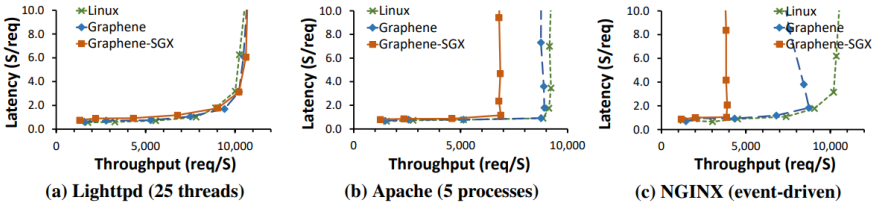

图 1: 三种应用 (Lighttpd、Apache、NGINX) 分别在 Linux、Graphene、Graphene-SGX 环境下的性能表现

能就会产生影响。**NGINZ** 服务器由事件驱动，因此 Graphene 的 shielding 层对数据的检查会有 overhead，因此性能更差。

不过，由于 Graphene-SGX 主要致力于将应用快速的部署到 SGX 提供的 enclave 中去，所以很多性能优化的工作都未集成。在已有快速部署已有应用这一特性的前提下，Graphene-SGX 的性能对先前工作来说也是很有竞争力的。

5 不足

无法防御一些攻击 Denial of service,sidechannels,controlled-channel attacks 这些攻击是目前所有的 SGX 平台都难以防御的，Graphene-SGX 同样无法处理。

不能保证可用性 像其他的利用 SGX 的平台一样，Graphene-SGX 可以保证代码在 enclave 内部的安全执行，但是无法保证 enclave 本身的可用性。例如，恶意的操作系统可以一直不返回到 enclave 中，或者在某些需要等待的操作中提前返回 enclave 使受保护的代码执行超出预期。

References

[1] Andrew Baumann, Marcus Peinado, and Galen Hunt. “Shielding Applications from an Untrusted Cloud with Haven”. In: *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*. Broomfield, CO: USENIX Association, Oct. 2014, pp. 267–283. ISBN:

978-1-931971-16-4. URL: https://www.usenix.org/conference/osdi14/technical-sessions/presentation/baumann.