使用硬件性能计数器实时检测基于缓存的侧信道攻击

要解决的问题是：在基于共享缓存的侧信道攻击完成之前发现它，以及时采取措施，比如在相同操作系统的情况下杀死可疑进程，或在跨VM场景中重定位虚拟机（将本来运行在同一个物理机上的两个虚拟机分开，将其中一个虚拟机重定位到另一个物理机上）。

文章介绍了三种方法：1.寻找受害程序和监视程序之间的相关性；2.基于异常检测的方法；3.基于神经网络的方法。它们可以组合使用，也可以分开使用，来实现在运行时检测基于缓存的侧信道攻击。使用这三种方法唯一的要求是硬件性能计数器的可用性。

## 针对RSA AES和ECDSA的侧信道攻击

## 硬件性能计数器

硬件性能计数器（HPC）：是现代微处理器配备的一类专用寄存器，用于存储和CPU相关的各种事件的数据，比如时钟周期，高速缓存是否命中，分支是否命中等。最原始的目的是为了软件调试和系统性能优化。在此文中，HPC被用来以很少的开销收集一个或多个进程的预测数据。在Linux中，可以通过名为perf的命令行工具为HPC提供交互式界面。

但是由于perf的分辨率有限，即两次连续采样之间的最小间隔为100ms。所以作者又开发了一个名为quickhpc的自定义实用程序，可以以普通用户级进程运行，但是需要和被监视的进程一样的特权（例如，要监视的进程如果以root身份运行，那quickhpc也必须以root身份运行）。该实用程序提供了perf-stat（*此程序通过概括精简的方式提供被调试程序运行的整体情况和汇总数据*）功能的一部分，但有一些改进。它的分辨率比较高，由于使用了性能应用程序编程接口（*PAPI，这是一组用于访问处理器硬件性能计数器的本地接口，用户可以利用它获得程序运行过程中产生的各种处理器事件的发生次数，进而将这些数据作为评估和分析程序性能的可靠依据，有助于更准确地发现程序性能问题的根源*），其分辨率可以达到3微妙。运行quickhpc时，可以指定最大采样数和两个采样之间的间隔（以微秒为单位）。但是quickhpc的分辨率会受到系统的负载，调度策略，被监视的进程等因素的影响。

## 异常检测

异常检测，用于在未被标记的数据集中查找异常值。假设对于数据集中的每个实例或样本都存在一组特征，我们可以根据这些特征来对数据集分类。用来表示第i个样本的第j个特征。异常检测的目标是检索大量的被认为是良性的样本，为这些样本找到一个合适的概率模型。一般会假设每个特征均符合高斯分布，且相对于所有样本中该特征值均具有均值和方差。因此为特征找到一个模型就意味着找到数学期望和方差满足正态分布（）。找到这样一些值之后，就相当于建立了一个模型，一旦有了新的样本，可以直接计算该样本的概率密度函数：



得到这个新样本属于之前建立的模型的概率。将这个函数返回值与阈值进行比较，来判断新样本是否属于异常。其中阈值是通过在包含已知异常的数据集上测试模型来确定的，也就是为数据集中的每一个样本数据按照已建立的模型计算它的第j个特征的概率密度函数，得到函数的返回值，然后根据这些返回值确定一个阈值，使得我们可以根据这个阈值可以将样本中的异常和正常数据区分开。

在实验中，将良性程序视为异常，监视程序视为正常，尝试为每种监视程序建立拟合模型。

## 监督学习和神经网络

* 神经网络：

神经网络是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的输出函数，称为激励函数。每两个节点间的连接都代表一个对于通过该连接信号的加权值，称之为权重。网络的输出则依网络的连接方式、权重值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近，也可能是对一种逻辑策略的表达。

即：

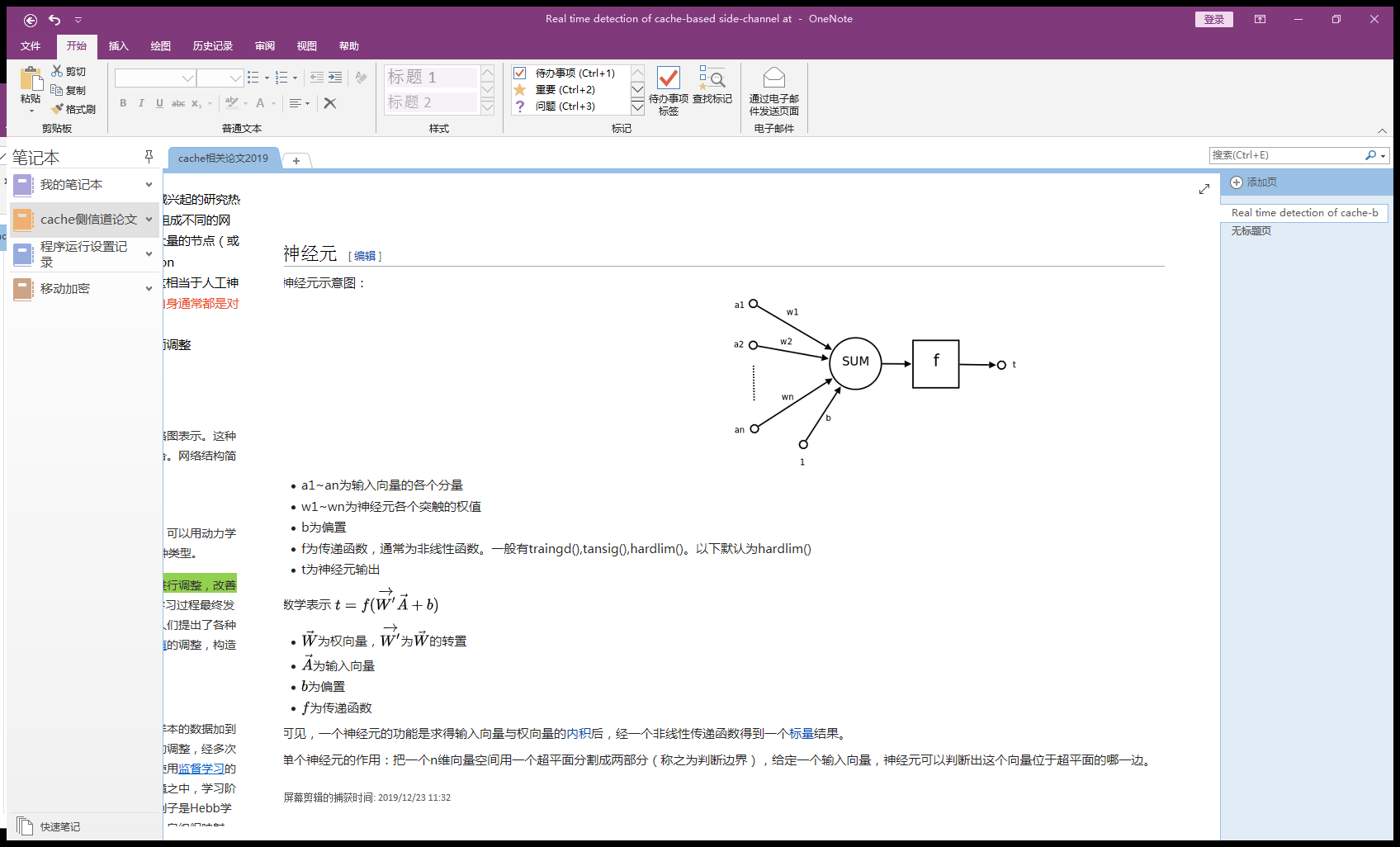
结构——神经网络的拓扑结构。

激励函数——节点。

权重——节点之间的连接

学习规则——网络中的权重如何随着时间推进而调整

单个神经元结构：



可以认为a1~an为输入向量的各个分量，w1~wn为神经元各个突触的权重，b为偏置，f为传递函数，通常为非线性函数；t为输出，是一个标量结果。

数学表示为：( (的转置)

由此，单个神经元的作用可以理解为把一个n维向量空间用一个超平面分割成两部分，给定一个输入向量，神经元可以判断出这个向量位于超平面的哪一边。

神经网络结构：



为了训练神经网络，前馈传播（参数从输入层向输出层单向传播）之后是反向传播。在此步骤中，将计算预测误差。然后，通过梯度下降算法利用这种误差来调整隐藏层中神经元的权重，以提高使用模型预测的准确性。

* 监督学习

根据学习环境不同，神经网络的学习方式可分为监督学习和非监督学习。

在[监督学习](https://baike.baidu.com/item/%E7%9B%91%E7%9D%A3%E5%AD%A6%E4%B9%A0)中，将训练样本的数据加到网络输入端，同时将相应的期望输出与网络输出相比较，得到[误差](https://baike.baidu.com/item/%E8%AF%AF%E5%B7%AE)信号，以此控制权值的调整，经过多次训练后收敛到一个确定的权值。当样本情况发生变化时，经过学习可以修改权值以适应新的环境。监督学习的主要目标是根据已知的已标记的样本构建一个分类器。训练阶段完成后，可以为分类器提供未被标注的单个特征向量，分类器应该能够预测特征向量属于哪类实体。监督学习中最难的任务是找到能够很好地描述某个类的特征。

监督学习和异常检测的区别：虽然它们都是分类，但是异常检测中一般样本中正常的数目会比异常样本的数目多很多，而且异常检测只能区分一个样本是否属于某一个主要的类别，而不能区分多个类别。但是在监督学习中，属于不同种类的样本的数量越平衡越好，而且监督学习可以区分多个类别。

在非监督学习中，事先不给定标准样本，直接将网络置于环境之中，学习阶段与工作阶段成为一体。非监督学习是在非标记数据中找到一个模型。

文中用来评估神经网络和异常检测系统执行情况的指标是F分数。这种度量方式比使用准确率（即计算所有预测中的正确预测所占的比例）来度量更可靠，因为它不受某些类别包含的样本数量比其他类别包含的样本数量更多的数据集的影响。

* F分数：是准确率（Precision）和召回率（Recall）的加权调和平均值：

,其中β为参数，P为准确率，R为召回率。当β=1时，称为F1-score，这时，准确率和召回率都很重要，权重相同。在计算时，如果认为准确率更重要些，那就调整β的值小于1，如果认为召回率更重要些，那就调整β的值大于1。F分较高时说明分类模型的性能比较好。

对于准确率和召回率的定义，首先先定义四种分类情况：

TP（True positives）：正类判断为正类 （属于此类的判断为属于此类）

FP（False positives）:负类误判断为正类（存伪，即将不属于此类的样本误判为属于此类）

FN（False negatives）：正类误判断为负类（去真，即属于此类的样本没有判断出来，误以为它不属于此类）

TN（True negatives）:负类判断为负类 （不属于此类的判断为不属于此类）

准确率=正类/（正类+存伪）

召回率=正类/（正类+去真）

## 检测一个监视进程

第一种是基于相关性的检测方法，后面两种是基于机器学习的检测方法。

* 1. 基于相关性的方法

这种方法是通过分析quickhpc收集的数据来找到受害者和间谍之间的相关性，使用的判断相关性的指标是随时间推移L3缓存的访问次数。因为监视程序和受害程序大部分时间都花在一个循环中，受害程序一般会在循环中计算某个函数或者访问某个变量，而监视程序在循环中刷新和重载其监视的地址，所以在循环中，它们都会定期访问可能在缓存中的数据，因此监视程序和受害程序在某个时间段内访问L3缓存的次数会很相似，以此来判断一个进程是否是监视进程。为此，文中还测试了三个Apache Web服务器上的良性进程访问L3缓存的次数（因为这种攻击主要针对服务器），随着时间的推移，它们访问L3缓存的次数和受害进程（比如ECDSA的蒙哥马利阶梯算法中for循环访问L3缓存的次数）相差很多，而间谍程序访问L3缓存的次数和受害程序访问L3的次数基本是一致的。这证明了这种指标是可用的。

这种方法在实际情况中，由于通常是无法知道在进程中何时发生了这种攻击，所以必须持续监视潜在的受害进程，并分别监视系统产生的每个进程。

开销：由于几毫秒的计算就可以以较高的精确度确定两个进程之间的相关性，所以监视阶段不会影响程序的整体性能。此外进行实验的时候由监视工具造成的开销可以忽略不计。

* 2. 基于异常检测的方法

由于可能存在误报，而且某些监视进程还可以通过故意创造噪音来逃避相关性检测，所以需要使用机器学习的方法来分析此类行为，并且减少误报，提高检测的可信度。

在各种试验后通过分析quickhpc获取的样本的每个特征集达到的F分，选择总指令数，总CPU周期，L2缓存命中数，L3缓存未命中数，L3缓存总访问数作为监视的特征集。

使用异常检测时，将来自监视进程的数据样本视为正常样本，其他进程的数据样本视为异常样本。通过使用一些已知的监视进程的样本数据，训练模型，直到找到最佳的阈值，训练阶段分为三步：

1. 为每个特征集计算期望和方差；
2. 为每个样本x计算总的概率密度函数值p（x）=,并找到一个阈值，使得如果x是一个异常数据样本，则p（x）小于此阈值。其中阈值的最佳值是根据每次迭代时在交叉验证集上达到的F分来选择的。
3. 在包含异常的数据集上测试p（x）并验证是否可以识别此类异常。

* 3. 基于监督学习的方法

在监督学习的情况下，quickhpc收集的原始数据先由一组脚本处理，合并到一个数据集中，再送入神经网络，输出代表两类：恶意进程和良性进程。在训练集中将受害进程标记为良性。受害进程的样本的存在对于网络区分行为非常相似但不属于同一类别的两个进程很有用。

## 实验结果

基于相关性的方法：

相关性及两个进程之间存在相关性的可信度的计算方法：（cov表示计算方差）



可信度越高，说明两个样本数据集之间存在相关性的可能性越大。样本的数量越高，获得良好的可信度的概率就越大。而且良性进程与受害进程之间相关性的最大可信度也要比监视进程与受害进程之间相关性的最小置信度低近一个数量级。在AES和ECDSA易受攻击的部分中执行时间最快的是蒙哥马利阶梯算法中的循环，最多需要2.8ms，而对具有500个样本的数据集执行算法8的平均时间为0.35ms，这样仍然有2.45ms的时间采取适当的措施。

基于异常检测的方法：

用时最短：对于100个样本的预测，平均只需要0.2ms。但是受噪声影响最大。

基于神经网络的方法：

用时较长，预测100个样本平均需要0.64ms。但是其F分比较高。

## 一种改进的攻击

这里使用了由Irazoqui等人修改的AES监视程序。由于相关性检测利用的是对缓存的访问的次数，所以监视进程可以在监视的主循环中再访问一组随机生成的数目的地址，因此就有了随机次数的cache的命中或未命中。这虽然会导致执行时间的延长，但是不会影响攻击的成功。此时可信度的范围明显减小了。

当使用神经网络和异常检测的方法来检测这种改进的监视程序的时候，神经网络得到的最大的F分为0.98，异常检测是0.79，类似于未经修改的AES监视程序，甚至比未经修改的监视程序的F分还要稍高一些，这说明新加入的行为使得该进程更加突出，使用基于机器学习的方法时，检测会变得更加容易。