无线传感器网络中的蠕虫传播节点检测模型

在第5章和第6章中，我们提出了静态和移动复制节点检测方案来检测和移除传感器网络中的静态和移动复制节点。然而，攻击者需要拥有与他想要使用的副本数量一样多的传感器节点，由于购买了许多传感器，导致了巨大的成本。为了通过复制节点攻击节省成本，攻击者可能试图通过捕获几个节点并让捕获的节点通过网络传播蠕虫病毒来大范围感染传感器节点。由于攻击者可以通过蠕虫病毒攻击整个传感器网络，因此蠕虫恶意节点非常危险，需要尽快被检测和移除。为了满足这一需求，我们提出了一种传感器网络中的蠕虫传播检测方案。该方案的核心思想是利用蠕虫传播节点的通信模式不同于正常节点的通信模式，结合SPRT序列化检测模型解决蠕虫传播检测问题。传感器网络中的蠕虫需要一长串数据包，依次逐跳传播到每个新感染的节点。因此，我们让检测器节点观察可能从它们的局部区域取样的通信模式；一条蠕虫一跳一跳地传播数据包，会迅速建立正常通信中看不到的连接链。

概述：

捕获大量节点并对其进行物理破坏需要时间和精力，并且可能会使攻击者面临被发现的风险，对于攻击者来说，更好的选择是仅仅捕获并破坏几个节点，并用自我传播的恶意软件(即蠕虫)感染这些节点。通过简单地将这些受感染的节点重新引入网络，蠕虫可以迅速传播并敢让大量传感器节点，让攻击者控制网络的大部分操作。

1. 蠕虫节点的快速检测方案，在采样数据越少越快（采样数据的期望）的情况下能够检测到恶意节点，可以感染传感器网络节点的危害降到最低。
2. 要保证蠕虫节点的一个高检测率，将网络中的蠕虫节点找到清除
3. 要保证检测算法的一个误报率较低的一个情况，不能将网络中的大量节点误判为恶意节点。

我们提出的方案的主要优点在于借助基于偏向采样的SPRT实现快速蠕虫传播检测。此外，它对任何类型的蠕虫都有效，因为它检测到蠕虫传播与数据包复制一致。因此，即使攻击者在传感器网络中创建并传播零日和多态蠕虫，我们提出的方案仍能检测这些蠕虫的传播，而基于签名的蠕虫检测方案将难以检测它们。

模型的假设条件：

在这一节中我们讨论蠕虫传播检测方案下网络结构和攻击者拥有的条件。

我们假设无线传感器网络是静态网络，其中传感器节点的位置在放置在网络域上后不会改变。我们还假设双向通信链路，通过该链路通信的传感器节点能够同时彼此监听。

我们假设攻击者将一组受危害的节点设置为蠕虫的发起者，并让它们在网络上传播蠕虫，从而在没有被检测到的情况下尽可能多地控制网络。由于流行病模型为估计不同时间点的预期感染率和受感染节点数提供了一种有效的方法，因此它被广泛用于互联网中的蠕虫传播建模，因此我们采用它来模拟传感器网

络中的蠕虫传播。具体来说，我们利用了简单流行病模型[2]的离散时间版本，假设蠕虫传播模型是流行病病毒传播模型：



代表时刻网络中受感染的节点数量，代表网络中的总节点数量，代表传感器节点的成对感染率。在这个模型中，我们假设一个传感器只有两种状态:易感和已感染。所有传感器节点最初都处于易受感染状态，除了处于感染状的蠕虫发起者。一旦易感节点被蠕虫感染，它的状态就会变成已感染的。

我们假设攻击者采用逐跳蠕虫传播策略，其中感染节点将蠕虫传播到其相邻的易感节点。攻击者可以利用这种策略，因为他可以通过逐跳节点联系来快速感染许多敏感节点。

我们假设需要多个数据包来将自传播蠕虫注入传感器微粒。这个假设是非常合理的，因为在当前的传感器网络体系结构中，最大数据包大小是28字节，因此很难生成适合如此小的单个数据包的功能蠕虫。一个使用1k字节内存的自传播蠕虫在哈佛架构的传感器微尘中被实现和测试。当worm以其最小尺寸传输时，传输worm所需的数据包数量从50个到100个不等。这意味着，在最新版本的传感器蠕虫技术中，至少有50个数据包用于将蠕虫注入传感器微粒。

我们还假设蠕虫不会通过任何广播协议传播，在该协议中，每个节点在接收到消息时都会重播消息。这个假设是合理的，因为蠕虫不是通过广播协议传播的恶意代码，而是自传播的恶意代码。

1. 蠕虫节点检测模型描述(数据包的重新传送次数作为判决的门限值)

蠕虫节点检测模型的核心思想在于：蠕虫节点的传播模式区别于正常的通信包传播模式。我们假设在一个典型的传感器网络中，网络节点之间不会频繁地发送一长串的分组包，因为这样做会耗费传感器节点的大量能量，导致网络不稳定。

为了减少节点的漏报率和 误报率，我们需要根据数据包重新传播的测量次数动态地建立阈值

在传感器网络部署完毕之后，每个节点以的概率将自己选举为蠕虫传播模型的检测节点，这些节点不受蠕虫节点的感染。这些节点不接受任何数据但是可以进行监控，把概率值设小一点，那么不会影响网络中的正常通信。

|  |
| --- |
| 算法1.包预处理单元算法(PPU) |
| 输入：接收到的数据包packet |
| If(packet.destation==u&&packet.source==u’s neighbor) |
| Then broadcast(SID,DID) to u’s neighbor 以的概率发送给邻居节点 |
|  |

每个蠕虫监测节点获取从邻居节点获取的“通信模式”，然后进行蠕虫病毒的监测。

软件证明技术和蠕虫传播的通信模式的SPRT检测方案相结合。

软件证明技术的关键思想是通过检查程序代码的完整性来检测程序代码中被破坏的部分。如果节点通过证明确定其邻居已经破坏了程序代码，它将不再与该邻居通信。

7.4 安全性分析（根据算法来分析）

\

1. 异常节点的检测率 Detection Accuracy
2. 蠕虫传播异常在给定的模型下所受的限制
3. 异常节点检测的误报率

7.6 仿真环境搭建，仿真实验

1000个传感器节点，1000\*1000的范围内，一组传感器节点分为50个，分为20组，每个组节点和邻居节点之间的通信距离半径为50米，传感器节点的分布满足高斯分布的随机位置：



其中的代表的是每一个组节点头的位置, 代表的位置选取的方差。

我们在两种情况下评估我们的方案:良性和蠕虫。在良性情况下，节点只发送良性数据包。在蠕虫情况下，节点只发送蠕虫数据包。这种情况分离的主要原因是研究我们方案的精确蠕虫传播检测能力和误报率，而不会分别受到良性数据包和蠕虫数据包的影响。在良性情况下，我们假设良性流量是根据泊松分布生成的，因此良性数据包的到达间隔时间遵循具有速率参数λ的指数分布。每个传感器节点向**δ**个随机选择的目的地发送良性分组，使得它首先均匀随机地选择每个目的地的开始时隙，然后开始从选择的开始时隙向每个目的地发送分组。

代表的是包传送的比例，