文章编号:1004-2474(2014)06-01030-05

一种检测被捕获节点的基于异常的算法研究

王 骐[1,2],蔡子元[2],范慧璞[2]

(1. 湖北第二师范学院 物理与机电工程学院,湖北 武汉 430205;2. Florida State University,Tallahassee 32310)

摘 要:无线传感器网络在许多应用场合里需要采集较敏感的数据,因此安全问题至关重要。一旦传感器节点被捕获,且没有采取相应措施,节点的密钥信息易被泄露,攻击者完全可伪装成这些节点,向网络任意注入错误的信息,由此导致网络的安全性能急剧下降。提出了针对被捕获节点的一种基于异常的入侵检测算法,能有效识别无线传感器网络的被捕获节点。算法对传感器节点间关系进行抽象,采用传感器网络的事件驱动特性来确定某节点在固定时间间隔内是否在发生数据包,基站通过检测可疑节点的数据包发送时间的差异来加以确认。算法不依赖于任何被捕获节点如何行动和密谋的假设,能识别出偏离正常行为值的最大多数被捕获节点,而不会出现"假肯定"。

关键词:无线传感器网络;捕获;异常;入侵检测;算法;仿真

中图分类号:TN918.91;TP393.08 **文献标识码**:A

Research on an Anomaly-based Intrusion Detection Algorithm for Compromised Nodes in Wireless Sensor Networks

WANG Qi[1,2],CAI Ziyuan[2],FAN Huipu[2]

(1. College of Physics and Electromechanical Engineering, Hubei University of Education, Wuhan 430205, China;
2. Florida State University, Tallahassee 32310, USA)

Abstract: Wireless sensor network was widely used in many applications, and sensitive data was needed to collect, thus security was crucial. Once the sensor node was compromised, and did not take the appropriate measures, the node's key information was easily leaked. The attacker could masquerade as these nodes to inject arbitrarily erroneous information the network, which led to a sharp decline in network security. An anomaly-based intrusion detection algorithm for compromised nodes was proposed, which could effectively identify the compromised nodes in the wireless sensor networks. Algorithm abstracted the relationship between the sensor nodes, using Event-driven to determine if the node sent data packets in a fixed time interval. According to the time difference, the base station confirmed if suspicious node was really a compromised one. Algorithm did not rely on any assumption how the compromised node acted and conspire to each other, and could identify the overwhelming majority of the compromised nodes which deviated from the normal behavior, without "false positives".

Key words: wireless sensor networks; compromise; anomaly; intrusion detection; algorithm; simulation

0 引言

相比于传统的有线和无线网,低功率无线传感器网络能在较大的地理范围内以自适应的方式快速部署。这种特性使无线传感器网络特别适用于实时、大规模信息采集,事件监视的重要军事应用中,如针对敌方的目标追踪和战场监控,但这些应用也给网络安全提出了全新的挑战。特别是由于传感器节点往往部署在开放的环境中,易受到物理攻击[1]。一旦获取了节点的密钥信息,攻击者完全可伪装成这些节点,向网络任意注入错误信息。基本的加密安全机制,如认证和完整性保护等,都不能有效防范这些伪装攻击[2]。因此,如何准确识别被捕获节点,并将它们排除在网络外,是无线传感器网络入侵检测技术面临的一个重要问题。入侵检测是无线传感器网络的重要安全防护手段,入侵检测技术可分为特征检测和异常检测两种。特征检测通过一个已知的入侵用户图来标记入侵,其能准确和有效地检测出已知入侵,但不能检测新出现的攻击。异常检测

收稿日期:2013-12-21

基金项目:2012年湖北省教育厅科研计划基金资助项目(B20123102)

作者简介:王骐(1970-),男,湖北武汉人,副教授,博士,主要从事无线传感器网络安全、嵌入式系统应用的研究。目前在美国 Florida State University 作访问学者,主要从事无线传感器网络安全的合作研究。

通过监测明显异于正常行为活动来预测入侵,其不需要入侵检测的前期知识,且可检测到新的入侵;但其不能描述入侵的形式,且可能出现很高的差错率[8]。

1　基于异常的入侵检测技术

针对被捕获节点的处理方法,基于异常的入侵检测技术目前主要分为 3 类。

1.1　基于定位的异常检测

文献[4-5]中阐述了一种检测恶意信标节点的方式。作者推断,如果一个信标节点被捕获,那么它将不可避免地要发送具有错误位置信息的信标信号。这是因为恶意信标节点发送的位置信息和信标信号一定是被篡改的。网络中的信标节点具有节点 ID 号及密钥,从而能与网络内的其他信标节点(表面看起来仍是非信标节点)进行通信。恶意信标节点发送的预估位置信息不是以信标信号为依据,所以这些信息与真实值相比不同。一旦有效信标节点从恶意信标节点获取信标信号,那么该被捕获节点将会被检测出。由于被捕获信标节点难以获取相邻节点间每次通信的循环时间,因此,如果恶意节点采取基于本地信标信号的重放攻击,那么它就易被发现。

1.2　基于网络/邻居节点稳定性的异常检测

文献[6]阐述了一种基于传感器节点特性(如相邻节点的固定信息)的入侵检测系统。实验中网络间的通信采用多对一的方式,即节点将沿着相对固定的路径向某个固定节点发送信息。因此,整个网络运行期间并不需要节点用来识别相邻节点的 HELLO 洪泛数据包。假设网络部署后不会增加新节点,且节点是固定的。另外,节点的发送功率不会改变。网络内的每个节点能识别它的相邻节点,每个节点使用相同的硬件及运行相同的算法,节点上运行的时钟与其他节点不同步[7]。根据以上设定的稳定网络,节点知道能从相邻节点获得的信息。为了进一步运用这个概念,需要选择两个传感器网络的运行参数,用来存储相邻节点的信息[8]。选取的参数是[9]:

1) 数据包到达率,即单位时间内到达的数据包。

2) 平均接收功率。

算法定义了一个用来存储预定数量数据包的缓冲区。这些数据包用来计算“数据包到达率”,以及后续数据包的“接收功率”这两个指标的可接受值的范围。如果这 2 个参数的值不在持续更新的范围内,那么就认定网络发生了入侵。入侵检测系统依靠这些节点来通知其相邻节点,网络可能存在入侵者。如果某节点从其固定数量的相邻节点处获取入侵者信息,那么该节点将把这个可疑节点标注为被捕获节点。

1.3　基于软件的面向嵌入式设备的认证

检测被捕获节点的另一种不同方法是根据代码认证,来验证运行于传感器节点的实际程序代码。还有一种基于硬件的认证方法,这种硬件内含一个安全的协处理器,用来检测相关嵌入式设备存储器的内容。文献[10]介绍了一种基于软件的认证技术(SWATT),它是通过软件方式独立运行“代码认证算法”来实现的。

这些技术都是通过某种方式,在外部验证运行于嵌入式设备的代码的正确性,可信检验器是实现目标的关键因素。被捕获节点至少有一行代码与运行于正常节点的预期代码不同。检验器有“未被捕获传感器节点”存储器内容的副本。检验器向节点发送一种特定信息,作为伪随机数生成器的输入,来创建随机存储器地址[11]。给设备的每个存储器地址进行“校验和”检验。校验器在本地运行相同的校验步骤,来计算预期的校验和值,再将这个预期值与节点返回的值进行比较。

由于被捕获节点存储器的内容被改变,所以它必须确定由伪随机数生成器创建的每个存储器地址是否已发生改变。SWATT 技术目前已在无线传感器网络中使用,不过执行检测和运行验证步骤均需额外时间。作者在基于 AVR studio V4.0 的仿真器上进行了实验,结果表明,计算校验和的时间差异变得更突出,这是因为被访问的存储器地址的数量在增加。

文献[12]也对网络做了相同的假设,确定节点是否为被捕获节点的过程发生了变化。基站被假定为受新方,它在鉴定被捕获节点时起着很大作用。基站认为存在潜在的被捕获节点,采用代码认证方法来确认可能的威胁。SWATT 技术的思想是,一旦节点被捕获,那么它必将执行与正常节点不同的行为,这种思想在算法设计中是一种重要的概念[13]。

本文提出了针对被捕获节点的一种基于异常的

入侵检测算法,能有效识别无线传感器网络的被捕获节点。算法的核心部分是对传感器节点间关系进行监视的一种抽象,并依此为基础,设计了一种精确识别被捕获节点的推理算法,该算法不依赖于任何被捕获节点如何行动和密谋的假设,算法能够识别出偏离正常行为值的最大多数被捕获节点,而不会出现"假肯定"。

2 算法

算法的设计基于以下特性:

1) 网络部署后,节点状态保持不变,不会有新的节点加入网络。

2) 传感器节点的时钟不同步。

3) 基站的节点 ID=0。

4) 所有的传感器节点(基站除外)运行的硬软件相同。

表 1、2 为算法中用到的符号。

表 1 传感器节点的符号

符 号	含 义
node_idp	被接收的数据包的源节点 ID
arrival_time	基于系统时钟的数据包到达时间
arrival_time_buffer[N]	N 个节点数据包到达时间信息的存储缓冲区
max_buffer_packets	可存储的最大数据包数量
num_buffer_packets	当前数据包数量
high_value	数据包到达时间的最大允许值
low_value	数据包到达时间的最小允许值
transmission_time_buffer[X]	最后 X 个数据包传输时间的存储缓冲区
node_ids	传感器节点的 ID
node_idcompr	基站确认的被捕获节点的 ID
node_idvalid	基站确认的正常节点的 ID
compromised[]	基站确认的被捕获节点 ID 的存储缓冲区

表 2 基站节点的符号

符 号	含 义
node_ida	可疑的被捕获节点 ID
node_idb	发送警报的节点 ID
alert_buffer[Y]	节点 node_idb 最后发送的 Y 个警报信息的存储缓冲区

算法的设计包括两部分:

1) 初始化阶段,在这个阶段传感器节点开始和它们的邻居节点通信。当有数据包到达时,节点把数据包的 node_idp 与 arrival_time_buffer[N]的当前记录进行对照。如果没有发现此 node_idp,且缓冲区可存储的最大数据包数量(max_buffer_packets)尚未达到最大,那么将会创建一个新的 node_idp 节点记录。arrival_time_buffer[N]存储数据包的最大 arrival_time (high_value)及最小 arrival_time (low_value)。初始化阶段,节点获取 N 个邻居节点的这些参数值,网络越大初始化需要的时间也越长。网络中的每个节点还有另外一个缓冲区 transmission_time_buffer[X],存储它最后发送的 X 个数据包的传输时间。用伪代码表示的节点初始化过程为:

```
/ * 节点的初始化过程 * /
On {arrival of} packet
If(node_idp! =0) AND (node_idp NOT in compromised[])
    If node_idp already in arrival_time_buffer
    If num_buffer_packets<max_buffer_packets
    If arrival_time>high_value
    high_value=arrival_time
    Else If arrival_time<low_value
    Low_value=arrival_time
    End If
    Else
    If (arrival_time>high_value) OR
    (arrival_time<low_value)
    Send ALERT(node_idp, node_ids) to base station
    End If
    End If
    Else
    If max N not reached
    Add node_idp to arrival_time_buffer
    End If
    End If
    End If
```

2) 将被捕获节点引入网络。我们这样来定义被捕获节点:它执行的某些功能与正常节点运行的功能有所不同。本文将这些被捕获节点设置成执行选择传递攻击(selective forwarding attack)。检测的执行过程如下:如果节点 B 发送给节点 A 的数据包的到达时间不在节点 A 为节点 B 存储的"数据包到达时间的最大允许值(high_value)"与"数据包到达时间的最小允许值(low_value)"之间,那么节点

A 将向基站发送一个报警信息。然后基站向节点 B 发出请求,请求节点 B 向基站发送“最后 X 个数据包传输时间(transmission_time_buffer[X])”,用伪代码表示为:

/ * 收到某节点的报警信息后,基站向可疑节点请求发送

REQUESTION_TRANSMISSION_TIMES( ),其中 node_ida 为可疑的被捕获节点 ID,node_idb 为发送警报的节点 ID * /

On {arrival of} ALERT(node_ida, node_idb)

Send REQUESTION_TRANSMISSION_TIMES( ) to

node_ida

/ * 可疑被捕获节点接收到基站的

REQUEST_TRANSMISSION_TIMES( )请求后所采取的行为 * /

On {arrival of} REQUEST_TRANSMISSION_TIMES( )

Send TRANSMISSION_TIMES (transmission_time_buffer) to base station

接收到节点 B 发送的最后 X 个数据包传输时间(transmission_time_buffer[X])后,基站进行判别。如果基站发现这些时间并不连续,那么基站将把节点 B 标记为被捕获节点,然后基站向网络内发送一条广播消息,通知网络内的其他节点:目前网络内存在入侵节点 B。用伪代码表示的基站判别及判别后的执行过程如下:

/ * 基站接收到可疑节点上传的 TRANSMISSION_TIMES(transmission_time_buffer)后,判别该节点是否是被捕获节点 * /

On {arrival of}

TRANSMISSION_TIMES(transmission_time_buffer)

For i<size of packet_buffer

If transmission_time_buffer[i]! =transmission_time_buffer[i+1]

/ * 比较第 i 和 i+1 个数据包的发送时间 * /

Compromised node identified

Send broadcast

COMPROMISED_NODE_FOUND(node_ida)

End If

End For

If compromised node not identified

Send VALID_NODE_FOUND(node_idvalid) to

node_idb

End If

一旦某个传感器节点接收到这条广播消息,将会执行:

1) 清除掉数据包到达时间缓冲区内关于这个入侵节点 B 的所有数据包到达时间的记录;

2) 把入侵节点 B 的 ID 加入到被捕获节点的 ID 缓冲区(compromised[]),这样就终止了与这个被捕获节点的所有通信,用伪代码表示为:

/ * 节点接收到基站发送的“某节点是被捕获节点”的广播消息后,正常节点所采取的行为 * /

On {arrival of}

COMPROMISED_NODE_FOUND(node_idcompr)

Clear out any values in arrival_time_buffer for node_idcompr

Add node_idcompr to compromised[]

另一方面,如果基站通过分析 B 发送的最后 X 个数据包传输时间(transmission_time_buffer[X]),认为节点 B 的传输时间连续,那么基站将通知节点 A:节点 B 并不是一个被捕获节点。相应地,节点 A 将重新更新到达时间缓冲区(arrival_time_buffer[N])内节点 B 的“数据包到达时间的最大允许值(high_value)”与“数据包到达时间的最小允许值(low_value)”,用伪代码表示为:

/ * 基站确认某节点不是被捕获节点后,正常节点所采取的行为 * /

On {arrival of} VALID_NODE_FOUND(node_idvalid)

Update transmission_time_buffer values for node_idvalid

由于传感器节点的这些值不断更新,发往基站的报警信息数量不断减少,所以网络内的数据包数量也减少了。在算法的设计中,基站是网络内所有节点均认可的可信节点,所以基站有权最终确认某个节点是否为被捕获节点。因此,在算法设计中,总是假设基站在网络生存期内不会被捕获。实际应用中,基站比其他节点的处理速度更快,因为它要发送或接收所有与算法检测相关的数据包。

3 仿真及分析

运用 TOSSIM 对算法进行仿真，仿真过程中在不同运行情况下设定了不同的参数，观测不同的参数会产生哪些不同的影响。每个节点的相邻节点数量由(Z%10)+1 (其中 Z 为网络中节点的数量)产生。

实验表明，到达时间缓冲区的数据包数量特定值对仿真并没有很大影响，除非选取的数据包值很大。如果选取的数据包值很大，那么初始化所需时间将比预期的更长。因此，无论网络节点数有多少，每次仿真选取的可存储的最大数据包数量值为 10。在所有仿真情况下，存储于传输时间缓冲区的数据包数量始终保持为常数 5，仿真中所有正常节点每隔 3 s 发送一次数据包。由于被捕获节点的传输时间缓冲区值不连续，因此传输时间缓冲区的数据包数量值可小于 5。网络中被捕获节点数量等于网络节点数量与 10 的模。在使用基站的无线传感器网络中，相比于传感器节点，被捕获节点具有更强的计算能力和更大的存储能力，但在仿真实验中，并不能通过某种设定值或选项来产生一种功能更强大的正常节点。

图 1 为所有被捕获节点被引入网络并被检测到所需的平均时间。实验表明，网络节点数越少，检测所需时间也越少。这是因为，网络节点数越多，根据(Z%10)+1，被捕获节点数也越多，因此检测出所有被捕获节点所花的时间也越长。其次，网络节点数越多，产生的流量也越大，因此处理基站所有请求的时间也越长。尽管 TOSSIM 一次能模拟所有节点，但随着网络节点数的不断增多，发送数据包所花的时间明显增加。

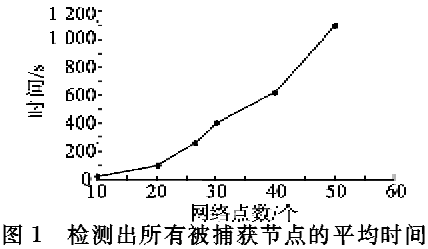

图 1 检测出所有被捕获节点的平均时间

图 2 为网络中被捕获节点发送数据包的平均数量。被捕获节点的数量取决于网络的规模，如具有 40 个节点的网络将会有 4 个被捕获节点。仿真实验中，具有 20 个节点的网络所发送的数据包数量为 1 825.67，被捕获节点发送数据包为 1.5，被捕获节点发送的数据包数量占全部数据包流量的 0.082 2%。这表明，相比于正常节点发送的数据包数量，在被检测出之前，被捕获节点发送数据包很少，因此该算法能有效阻止被捕获节点向网络内发送错误信息。一旦恶意节点被发现，任何与被捕获节点之间的通信将会被阻止，任何发往被捕获节点的数据包将会被忽略。

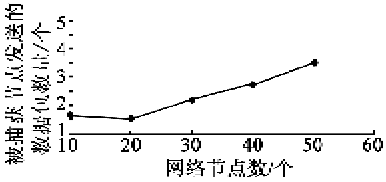

图 2 被捕获节点发送数据包的平均数量

“假肯定”通常是入侵检测系统面临的问题，指的是正常节点被认定为被捕获节点的情况。如果“假肯定”的节点增加，那么算法的效率就会显著降低。算法中，假定受信的基站自身不存在“假肯定”，并由基站处理“假肯定”问题。即一个正常节点不会被基站认定为被捕获节点。实验中，被基站阻止的“假肯定”节点的数量如图 3 所示。正常节点被“假肯定”存在多次重复，所以被基站阻止的“假肯定”节点的数量大于网络实际节点数量。

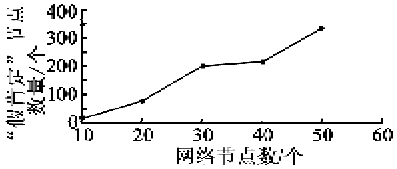

图 3 被基站阻止的“假肯定”节点的数量

4 结束语

本文阐述了一种检测无线传感器网络被捕获节点的算法，算法基于异常入侵检测技术。算法采用传感器网络的事件驱动特性来确定某节点在固定时间间隔内是否在发生数据包。基站通过检测可疑节点的数据包发送时间的差异来加以确认。算法的效率和准确率是最主要的 2 个方面，通过仿真，这 2 个性能参数均得到了验证。如果基站具有更强大的计算能力和存储能力，那么检测被捕获节点的时间会进一步缩短。

参考文献：

[1] LAW Y W, HAVINGA P J M. How to secure a wireless sensor network[C]//Vienna, Austria: Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005: 89-95.

(下转第 1038 页)