Towards an Intrusion Detection System for Pervasive Computing Environments

Pradeep Kannadiga & Mohammad Zulkernine
School of Computing
Queens University, Kingston
Ontario, Canada K7L 3N6
{pradeep, mzulker}@cs.queensu.ca

Sheikh I. Ahamed
Dept. of Math, Stat. and Com. Science
Marquette University PO Box 1881
Milwaukee, WI 53201-1881, USA
iq@mscs.mu.edu

Abstract

A pervasive computing environment consists of numerous casually accessible, frequently mobile, embedded, handheld or portable smart devices capable of sensing the environment around it and reacting intelligently to simplify user activities. These devices are distributed everywhere at office, homes, stores, classroom, and are often connected to ad-hoc network and the Internet providing access to any computing resource and services from anywhere and anytime. Access control mechanisms can sometimes fail to provide complete security to these pervasive computing devices as witnessed in infrastructure based networks. Often these pervasive computing devices are connected to infrastructure based networks like office LAN. This makes the pervasive computing devices also vulnerable to the same type of attacks as on infrastructure based networks. Hence, there arises the need for intrusion detection systems capable of operating both in infrastructure based networks and the ad-hoc networks of pervasive computing devices. Much of the research on intrusion detection has been carried out in the field of infrastructure based networks. The addition of pervasive computing devices to infrastructure based networks makes the problem of intrusion detection even harder. In this paper, the challenges and characteristics of intrusion detection in pervasive computing devices are discussed. The paper also describes the detailed architecture of an intrusion detection system using mobile agents for a network environment made up of infrastructure based network and pervasive computing devices.

Keywords. Pervasive computing security, intrusion detection, and mobile agents.

1. Introduction

In the recent years, pervasive computing [1] applications have grown tremendously due to the recent developments in portable low-cost lightweight devices and emergent short range, and low power wireless communication networks. The pervasive computing environment consists of numerous small, casually accessible, frequently mobile, embedded, handheld or portable smart devices connected to an ad-hoc network structure [2, 3]. These devices are capable of sensing the environment around it and reacting intelligently to simplify user activities. Pervasive computing provides a user access to computing resources and services from any location and at any point of time [1, 4]. Lightweight pervasive computing devices like Personal Digital Assistants (PDAs), mobile phones, and laptops can be carried everywhere by a user. These devices can be connected to each other and also to Internet by using wired or wireless technology. For example, wireless technology like Bluetooth [5] can provide LAN and Internet access to a handheld computer wirelessly through access points.

Security risks increase in a pervasive computing environment since a user can access resources and services from anywhere. A user with malicious intent can access, modify, or delete sensitive information present in hosts or make some of the computer services unavailable to the users. Access control mechanisms can fail to completely protect a device against all types of attacks resulting in information loss/theft. Moreover, the pervasive computing devices are also prone to large set of software attacks like the denial of service (DOS) or buffer overflow attacks that can make a device unavailable. These devices when connected to infrastructure based networks like office LAN, make them vulnerable to the attacks similar to that on infrastructure based networks. We need an intrusion detection system (IDS) to detect such attacks. Much of the research on intrusion detection have been carried out for traditional infrastructure based networks which comprise of computing nodes like computers, servers, switches, IP phones, routers. Addition of mobile, handheld, and portable pervasive computing nodes into existing fixed infrastructure based networks increases the heterogeneity of the resulting

network, making the problem of intrusion detection even harder.

In this paper, we discuss the characteristics of intrusion detection for pervasive computing environments, followed by the detailed description of a mobile agent based IDS to be deployed in a pervasive computing environment. In our previous work [6], we developed a prototype of a mobile agent based distributed IDS with focus on infrastructure based networks. In this work, we extend the IDS for pervasive computing environments. In the infrastructure based network, the IDS uses a set of software entities called mobile agents that can move from one node to another node within a network, and perform the task of aggregation and correlation of the intrusion related data that it receives from another set of software entities called the static agents. The static agents are installed in every local monitored node of the network to detect signs of suspicious behavior in that node. However, in remote and small pervasive computing nodes like PDAs, static agents are not installed. Instead, only the mobile agents are used for intrusion detection. Mobile agents have the ability to move the execution state of resource intensive intrusion detection functions from handheld/portable pervasive computing nodes to other suitable nodes with sufficient memory and CPU speed. Mobile agents can operate autonomously and support heterogeneous platforms. The IDS discussed in this paper utilizes the above-mentioned beneficial features offered by mobile agent technology to perform intrusion detection for the pervasive computing environments.

The remainder of this paper is organized as follows. Section 2 presents other approaches related to the work done in this research. Section 3 describes the characteristics of IDSs for pervasive computing environments. Section 4 presents our approach for intrusion detection. Section 5 concludes the paper and discusses some future work.

2. Related Work

The eBiquity: research group [7, 8], views intrusion detection to be a "distributed and collaborative" process. Their research work focus on enabling secure access and service discovery in pervasive computing using a security infrastructure built upon Public Key Infrastructure (PKI) for user authentication, non-repudiation, and access control. Distributed trust management is used to complement existing PKI and role based access control security mechanisms. This solution provides a flexible security framework for pervasive computing environments. The eBiquity research projects do not discuss about intrusion detection in pervasive computing devices like PDAs.

In [9], agents for intrusion detection are placed in every node of the mobile ad-hoc network. These agents detect any anomaly in the node by using local audit traces and also communicate with agents of neighboring nodes to detect distributed attacks on the whole network. The focus in this work is more on intrusions in mobile ad-hoc network routing protocols such as route logic compromise and traffic pattern distortion. In our work, we detect a different set of software attacks like DOS, buffer overflow, and computer viruses in individual pervasive computing nodes instead of attacks on the network connecting these nodes which can be wired or wireless. We have used mobile agents in mobile wireless nodes that operate autonomously for intrusion detection instead of placing agents permanently in the nodes that collaboratively detect intrusion as discussed in [9].

The following work are on mobile agent based IDSs but closely related to infrastructure based networks. The Intrusion Detection Agent (IDA) system [10] consists of sensors running in every monitored host that report Marks Left by Suspected Intruder (MLSI) and a central manager responsible for dispatching tracing agents to the host whose sensor reports an MLSI. The tracing agents gather information related to intrusion from the sensors and send it to central manager for analysis. In our IDS, mobile agents that are similar to tracing agents used in IDA system, have dual functions: gathering intrusion related data from every host and analyzing the gathered data by itself instead of sending it to a central manager for data analysis. Mobile agents are employed to apply human immune system model for intrusion detection in [11]. This IDS works on anomaly based detection principle where each mobile agent travels to every host in the network to detect any deviation from the normal behavior of that host. The intelligent agents for intrusion detection project [12] have developed IDS using distributed multiple layers of lightweight intelligent mobile agents that apply data mining techniques to detect intrusions.

3. IDS Characteristics for Pervasive Computing Environments

In this section, we discuss in detail the security threats in pervasive computing environments that justify the need for an IDS. We also discuss the challenges and different approaches for intrusion detection in pervasive computing environments.

Privacy, trust, and availability are important in pervasive computing [4, 14]. Handheld devices like PDAs [13] can be used as storage of important and confidential information of their users. A stolen PDA can reveal such private information to others. Physical attacks like loss/theft, tampering of hardware are more common with pervasive computing devices. Software attacks like exploiting a flaw such as buffer overflow in application software or security loopholes, computer viruses, DOS attacks, wireless packet sniffing, and unauthorized access are also targeted against pervasive computing devices. For example: a) macro viruses infecting Microsoft office running in PDAs, Palm OS viruses, b) misconfigured Bluetooth PDA can allow any device to initiate connection to it due to ad-hoc communication between devices. The DOS attacks can make pervasive computing devices unavailable. The constant interaction between pervasive computing devices also needs increased trust. However, some of these devices may also operate in a hostile or untrusted environment. Much of the research on security related to pervasive computing carried until now have been on providing secure infrastructure based on trust management and access control mechanisms like role based access control, public key infrastructure, and biometrics [7, 8]. The access control mechanisms, encrypted communication lines, and trust management offer first layer of defense that can sometimes fail to protect a device completely against the software attacks discussed earlier in this section. Hence, we need an IDS to detect and respond to these attacks.

The intrusion detection techniques used for infrastructure based networks cannot be directly applied to pervasive computing environments. The reasons are as follows. First, there is no single point in the network where an IDS can be placed to monitor the network traffic since devices are distributed at different locations in a pervasive computing scenario. Fig. 1 shows an ad-hoc network of pervasive computing devices consisting of a laptop computer, PDA and mobile phone. All of these devices can be present at any location and connected to each other through wireless medium. Second, normal desktop computers or servers have enough computing resources for hosting IDS sensors that monitors the host or the network. Most of pervasive computing devices have limited computing resources like memory, processor speed, network bandwidth, and are mostly battery-powered devices. Hence, these devices are not capable of hosting a complete IDS sensor. Therefore, a pervasive computing environment requires a different approach for intrusion detection. Some of these are: 1) mining of pervasive databases to detect intrusion patterns; 2) lightweight software entities placed in every device to monitor intrusion attempts; and 3) intrusion detection circuitry built to detect intrusion and corrupted sensors. We propose the usage of lightweight mobile software entities called mobile agents for performing intrusion detection. The detailed description of this approach along with the architecture of the proposed IDS is discussed in the next section.

4. Our Approach

The presented IDS is designed by keeping in mind the notion of heterogeneity and scarcity of computing resources in a pervasive computing environment. The proposed IDS operates in a computing environment consisting of a local infrastructure based network (desktop computers, servers, and routers), which is connected to a remote ad-hoc network of lightweight handheld or portable pervasive computing devices (PDAs, laptop computers, and mobile phones). Fig. 1 shows the operating environment of the IDS made up of computers (Host1, Host2, and Host3) connected to local network, remote laptop computer (Rhost-A), and remote PDA (Rhost-B).

The proposed IDS is made up of static and mobile agents. The components are as follows: Static Agents (SA), Mobile Agents (MA), Mobile Agents Server (MAS), The Victim Host List (VHL), Mobile Hosts List (MHL), and Alerting Agent (AA). The architecture and the working environment of IDS are shown in Fig. 2.

An SA installed in every host of the LAN (Host1, Host2, and Host3), generates an event when a suspicious activity is detected. The SA sends events related to such behavior of the host to the MAS, which then creates an MA to handle the task of detecting intrusions based on such activities. The MAS component resides in a separate mobile agents server machine as shown in Fig. 1. The VHL stores the IP addresses of hosts of the LAN in which suspicious activity is detected. The MHL contains information of all the remote hosts that are currently connected to the local network as indicated by dotted lines in the Fig. 2. The MAs are of two types: Thick MA and Thin MA (see section 4.3 for the differences between thick and thin MAs.) Thick MAs are meant for local nodes (Host1, Host2, and Host3) while thin MAs are used for remote nodes (Rhost-A, Rhost-B). A thick MA gathers data related to intrusion from SAs running in those hosts listed in VHL, correlates and aggregates the gathered data, generates alerts on the detection of any attack, and finally returns to the

MAS. The solid lines with arrowhead in Fig. 2 shows the movement of MAs in the local network and its interaction with SAs installed in the hosts listed in the VHL. However, in remote (Rhost-A, Rhost-B) nodes, the SAs are not installed to avoid utilization of the limited resources like battery power, memory, and processor speed available in the remote nodes.

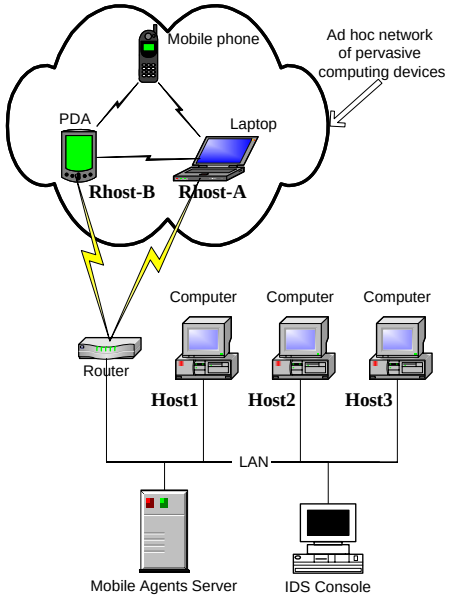

Fig 1: A typical pervasive computing environment with IDS.

The MAS dispatches a thin MA to every host listed in the MHL. The MA dispatched to remote node is responsible for detecting any suspicious activity in the remote node. The MA gathers audit trail left by the applications running in remote node for detecting any anomaly or intrusion at the remote node and carries it back to the MAS. The MA performs the analysis of the gathered intrusion data at the MAS and thus reduces the resource utilization in the remote nodes. The dashed lines with arrowhead originating from the MAS in Fig. 2 show the movement of MAs through wireless medium to the nodes in the remote network. An AA module receives generated alerts from MAs and displays the alerts to the security administrator. The AA component resides in the computer that hosts the IDS console as shown in Fig. 1. After an MA is dispatched the MAS waits for that MA to return in order to prevent the MAS from dispatching the MA for the same task twice. The components of the IDS are described in detail in the following subsections.

4.1. Static Agent (SA)

Every monitored host in the LAN has a static agent component. Static agents act like host monitors generating events to indicate attacks, and these events are sent to MAS. Each event carries information of the probable type of attack. For example, an SA identifies failed password guessing attempts as a suspicious activity, and an event is generated to check for doorknob-rattling attack [15]. An SA is a multithreaded program where each thread monitors the host for different classes of attacks. These threads run at every administrator configured time interval to check the log files for any trace of attack. The generated events are sent as a message to a remote object in the MAS whenever a trace of an attack is detected. An SA is responsible for triggering the movement of an MA within the local network. The threads of an SA run at a lower priority level and hence do not cause slowdown in the execution of other programs at the host system. The SA contains objects specific to attack(s) that are responsible for parsing the log files, checking for intrusion related data pattern in log files, separating data related to the attack from the rest of the data, and formatting the data as required by an MA.

4.2. MA Server (MAS)

The MAS is a repository of all the MAs used by the IDS. It is responsible for dispatching the MAs to target nodes both local and remote. An MAS component contains two other components MHL and VHL that decide the itinerary of an MA. In the local network, the MAS decide about the MA that has to be dispatched according to the attack event generated by SA. The object request broker module within the MAS contains objects that can receive messages from SA(s) about the detection of suspicious activities in some of the host(s) of the network. These objects are then responsible for creating an MA and sending it to the victim host(s).

Victim Host List (VHL) maintains separate lists to store the IP addresses of all the hosts that are subjected to the same types of attacks. For example, all the hosts subjected to doorknob rattling attack are maintained as a separate list in VHL. VHL provides the itinerary for the movement of an MA within the local network. When MAS receives an event message from an SA, the IP address of that SA host is added to the respective list in the VHL. All the MAs originate and return finally to this component.

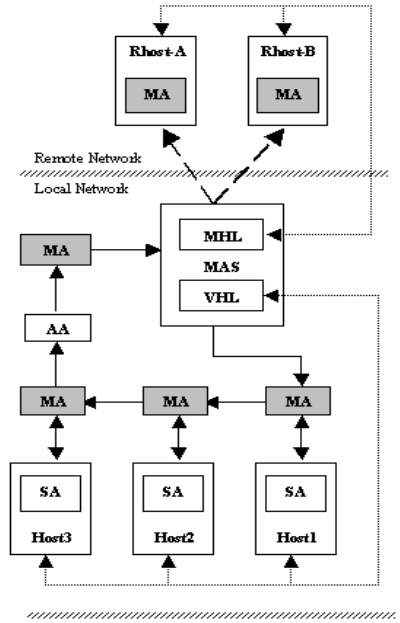

Fig 2: The IDS architecture.

The MHL component within the MAS keeps track of all the remote hosts currently connected to the network. It stores the time of connection, device type, and IP address information of the remote node. The device type information helps in deciding if a thick or thin MA has to be dispatched. The MAS dispatches a thin MA to every remote node at regular interval of time. A security administrator depending on the device type can configure the interval at which the thin MAs are dispatched to remote nodes. These thin MAs use MAS for performing analysis of the audit data gathered from visited remote nodes.

4.3. Mobile Agent (MA)

Mobile agents are responsible for collecting evidences of an attack from all the attacked hosts and for further analysis of the gathered data. The advantage of using an MA is that it can function autonomously after it is dispatched without any control from other components. An MA can operate even if the network is disconnected. This property suits pervasive computing nodes that are usually intermittently connected to Internet or office LAN. The MAs are of two types: thick MA and thin MA.

Thick MAs are for nodes with sufficient resources to support a mobile agent platform. In the local network, the thick MAs are used since the nodes are usually computers or servers capable of hosting an agent platform. The thick MAs take data from all the SAs running in the hosts listed in the VHL, correlate and aggregate the data, and generate any alerts on the detection of any intrusion activity in these hosts.

Thin MAs have low system footprint and capable of running in small resource constrained nodes. Thin MAs are developed considering nodes like PDAs and mobile phones that have OS, memory and processor speed which cannot support a complete mobile agent platform. For remote networks, a separate thin MA is dispatched for every node present in the MHL list.

Thin MAs contain only the code to parse audit trail left by application software, gather intrusion related data, format them, and move the data to MAS. The data gathered by a thin MA is analyzed at the server hosting the MAS component. Thus it relieves the remote nodes with limited memory and CPU speed from resource intensive intrusion data analysis. Alerts are generated on the detection of any traces of attack in the analyzed data.

4.4. Alerting Agent (AA)

An alerting agent receives alerts generated from both thick and thin MAs. It stores the information gathered from an MA in a separate log file for further analysis. An AA consists of an IDS console that can be used by the security administrator to view alerts generated from the MAs. The AA is responsible for preventing multiple alerts being generated for the same attack, and for suitably formatting the data to provide complete information about an attack to the administrator.

4.5. Implementation of MAs

Voyager [19] is used as the mobile agent platform for implementing the thick MAs, where each thick MA contains code only for detecting a specific type of attack resulting in small size code. Voyager supports only Desktop computers and is not suitable for implementing the thin MAs. A thin MA utilizes fewer resources in the remote node compared to a thick MA. The widely used open source mobile agent platforms [5, 16, 17, 18] currently available for handheld/portable computing devices are JADE-LEAP and MicroFIPA-OS. MicroFIPA-OS is mainly targeted at "Compaq IPAQ handheld computers". JADE-LEAP supports large variety of devices and has the smallest system footprint. It currently supports PDAs, Java-enabled mobile phones in addition to desktop computers. JADE-LEAP supports TCP/IP over GSM and IEEE 802.11 Wireless LAN protocols. Therefore, JADE-LEAP is an intuitive choice for mobile agent platform for the pervasive computing devices of our IDS.

5. Conclusions and Future Work

We have presented the architecture and operation of an IDS based on mobile agents technology for a network of pervasive computing devices connected to infrastructure based network. This IDS uses static and mobile intrusion detection agents. The static agents are confined to local devices but the mobile agents are used at both local (thick MA) and remote mobile nodes (thin MA.) The main advantage of this approach is that mobile agents can move resource intensive intrusion detection functions from nodes with limited computing resources to other suitable nodes with sufficient memory and CPU speed. The IDS can be easily extended by adding new MAs for detecting new attacks, or the existing MAs can be modified for better detection capability, resulting in a highly modular and extendable architecture. A prototype IDS is being implemented using JADE-LEAP on Dell Axim X30 Pocket PCs and Voyager on desktop computers. In future, we will extend this prototype to support variety of other devices like Palm OS-based PDAs and Java enabled mobile phones.

6. References

[1] M. Weiser, "Some Computer Science Problems in Ubiquitous Computing", *Communications of the ACM*, Vol. 36, No. 7, pp. 75-84, July 1993.

[2] S. S. Yau, F. Karim, Y. Wang, B. Wang, and S.Gupta, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing", *IEEE Pervasive Computing, joint special issue with IEEE Personal Communications*, pp.33-40, July-September 2002.

[3] S. S. Yau, S. Gupta, F. Karim, S. Ahamed, Y. Wang, and B. Wang, "A Smart Classroom for Enhancing Collaborative Learning Using Pervasive Computing Technology", *Proc. of the 6th WFEO World Congress on Engineering Education & 2nd ASEE International Colloquium on Engineering Education (ASEE2003)*, Nashville, Tennessee, USA, June 2003.

[4] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, IEEE Computer press, August 2001.

[5] Alf Inge Wang, Carl-Fredrik Sorensen and Eva Indal, "A Mobile Agent Architecture for Heterogeneous Devices", *Proceedings of the 3rd IASTED International Conference on Wireless and Optical Communications (WOC 2003)*, Banff, Canada, July 2003.

[6] P. Kannadiga and M. Zulkernine, "DIDMA: A Distributed Intrusion Detection System Using Mobile Agents", *Submitted to International Workshop on Systems and Network Security (SNS2005)*, Colorado, USA, 2004.

[7] Jeffrey L Undercoffer *et al*., "A Secure Infrastructure for Service Discovery and Access in Pervasive Computing", *ACM Monet: Special Issue on Security in Mobile Computing Environments*, October 2003.

[8] Lalana Kagal *et al*., "A Security Architecture Based on Trust Management for Pervasive Computing Systems", *Proceedings of Grace Hopper Celebration of Women in Computing*, October 2002.

[9] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks", *ACM Wireless Networks Journal*, 9(5): 545-556, September 2003.

[10] M. Asaka, S. Okazawa, A. Taguchi, and S. Goto, "A Method of Tracing Intruders by Use of Mobile Agents", *INET'99*, San Jose, USA, June 1999.

[11] N. Foukia, J. Hulaas, and J. Harms, "Intrusion Detection with Mobile Agents", *Proceedings of the 11th Annual Internet Society Conference (INET 2001)*, Stockholm, Sweden, June 2001.

[12] G. Helmer, S. K. Johnny, Wong, V. Honavar, and Les Miller, "Intelligent Agents for Intrusion Detection", *Proceedings of the IEEE Information Technology Conference*, NY, USA, pp. 121-124, September 1998.

[13] David B. Rankin, "Handheld Computer Security", East Carolina University, July 2004.

[14] Kumar Ranganathan, "Trustworthy Pervasive Computing: The Hard Security Problems", *Proceedings of the 1st IEEE Intl. Workshop on Pervasive Computing and Communication Security*, Orlando, Florida, March 2004.

[15] Ko, D. Frincke, T. Goan, L. T. Heberlein, K. Levitt, B. Mukherjee, and C. Wee, "Analysis of an Algorithm for Distributed Recognition and Accountability", *Proceedings of the first ACM Conference on Computer and Communication Security.* Fairfax, VA, Nov. 1993.

[16] Mikko Laukkanen, "Agents on Mobile Devices", http://www.cs.uta.fi/kurssit/AgO/ago5-print.pdf.

[17] Bergenti, Federico, Poggi, and Agostino, "LEAP: A FIPA Platform for Handheld and Mobile Devices", 2001. http://leap.crm-paris.com/public/docs/ATAL2001.pdf.

[18] TILAB, JADE (Java Agent Development Framework), 2002. http://sharon.cselt.it/projects/jade.

[19] Recursion Software Inc, "Voyager ORB Developer's Guide", www.objectspace.com, http://www.ifi.unizh.ch/ddis/staff/vorburg/doc/Orb/index.htm.