IMoViS: a system for mobile visualization of intrusion detection data

Andrea Sanna and *Claudio Fornaro*

Dipartimento di Automatica e Informatica,
Politecnico di Torino, corso Duca degli Abruzzi 24,
I-10129 Torino (Italy)

January 5, 2004

Preferred address for correspondence:

Prof. Andrea Sanna

Dipartimento di Automatica e Informatica,

Politecnico di Torino, corso Duca degli Abruzzi 24,

I-10129 Torino (Italy)

Tel.: +39-11-564.7035

FAX: +39-11-564.7099

e-mail: andrea.sanna@polito.it

IMoViS: a system for mobile visualization of intrusion detection data

Keywords: intrusion detection, information visualization, mobile devices, PDA.

Abstract

Mobile devices such as PDAs allow a sort of ubiquitous access to the Internet. This can be of great value for all disciplines where information has to be conveyed to the user in "real time" independently of his/her physical location. Intrusion detection applications can take advantage of the employment of mobile devices by allowing a constant monitoring of the state of a computer system.

This paper proposes an integrated framework to visualize intrusion detection data on PDAs. The Snort ID system is used to detect attacks and intrusions and to store the gathered information into a database. The information is processed by a software called Guardian that produces the actual data to be fed to the visualization application. The proposed architecture is tailored for monitoring large buildings by organizing spatial data information in a hierarchical way. The user can discover and manage attacks/intrusions at the top level of the hierarchy (the entire building) as well as at the leaf level (the single machine placed into a room) where detailed information about the attack can be obtained.

1 Introduction

Intrusion detection applications often produce a large amount of data. The visualization of this information is a key task in order to allow the user to effectively detect attacks and intrusions. Information visualization is an important sub-discipline within the field of scientific visualization and focuses on visual mechanisms designed to communicate clearly to the user the structure of information and *facilitate* the access to large data repositories. Information visualization enables people to deal

with all of this information by taking advantage of visual perception capabilities of the human being. By presenting information in a more graphically oriented fashion, it is possible for the human brain to take advantage of its perceptual system in the initial information acquiring, rather than immediately relying entirely on the cognitive system. Some of the most important papers in the field are collected in [1]. Information visualization algorithms require the merging of data visualization methods, computer graphics solutions, and graphical interfaces design.

A new challenge in information visualization is the employment of Personal Digital Assistant (PDA) devices. PDAs were originally designed as personal organizers and their employment was limited for a number of reasons, among them: small screen size, low computational resources, limited wireless communication bandwidth, reduced interaction capability. But today's PDAs are efficient pocket computers and thus can be effectively used for remote monitoring purposes.

This paper proposes an integrated architecture by which a security manager can remotely monitor large buildings for computer intrusion attempts, the front-end used is just a PDA. The system is composed by two different parts:

- the intrusion detection and information collection system;
- the visualization interface.

Information collection is achieved by monitoring network traffic of the LAN under control. The program used intercepts all the network traffic and scans it for traces of a possible attack.

For the Intrusion Detection part, an architecture based on existing software has been defined. Its purpose is to set some preprocessing steps for the information data, mainly cleaning and reduction, in order to meet the limited computation and communication resources of mobile devices.

From the visualization point of view, this paper presents a graphical interface designed for PDAs. Data related to the building are organized hierarchically, this allows the user to see a global view as well as a detailed information concerning every machine located in each office/room of the building.

Moreover, a tool for designing building maps is proposed. An arbitrary number of machines can be placed in every office and a set of information such as IP address, operating system, user name, office telephone number, and so on is associated to every machine.

The paper is organized as follows: Section 2 reviews both main concepts of intrusion detection and some examples of visualization on PDAs, while Section 3 explains in the details the proposed architecture and shows how intrusions are displayed on a PDA. Finally, Section 4 presents some remarks concerning performance issues of the proposed framework.

2 Background

2.1 Intrusion Detection Systems

To hinder the attacks performed against computer systems, information and security professionals turn to *Intrusion Detection Systems* (IDSs) to set up an active defense-in-depth strategy. A *firewall* is an essential and important part of network security but it does not have the ability to detect hostile intent. IDSs are classified in Host IDSs and Network IDSs. While the former are installed directly on the machine to be monitored and are intended just for that machine, the latter perform surveillance for the whole network. The most common schema for a NIDS is composed by at least one sensor that intercepts the network traffic and an analysis engine. Alerting and attacks analysis may be handled by different machines. An analysis engine looks at packet protocol flags, source and destination addresses, sequential numbers, and application payload such as email messages or web requests. Common attack signatures consist of strings to search for in the payload or network packet parameters. In addition, analysis may be applied to the whole TCP connection rather than individual packets or even include correlation of the connection to those occurred earlier or elsewhere on the network. The advantage of a NIDS is the ability to protect an entire network with a single machine, in a transparent way with respect to network hosts, with no impact on the network architecture and performance, detecting not only actual attacks, but even potential ones. Traffic wiretapping (sniffing) is possible because of

the way data are transmitted over a LAN. Unencrypted data are split in packets (frames) and each one is directed to a particular NIC which is identified by an address (a 48-bit number) called the MAC Address. No two NICs are manufactured with the same MAC Address (however, some NICs allow to change it). When a packet travels the LAN, all the NICs that see it read the embedded destination MAC Address, but only the NIC whose MAC matches the one in the packet reads the whole packet and forwards it to the machine network protocol stack (e.g. TCP/IP) to be processed. On the contrary, NIDS sensors use a NIC set up in *promiscuous mode*, so that they read all the network traffic, all the packets, independently of the destination MAC Address. The NIDS can thus have a big picture of the entire network and is able to recognize attacks conducted to every machine connected to the LAN.

2.2 Visualization on PDA devices

PDAs and more in general mobile devices proved to be very effective tools for a large range of disciplines. For remote visualization a survey can be found in [2]. This section briefly reviews main areas where mobile devices (and among them PDAs) are employed.

2D and in particular 3D graphics can be computed directly on PDAs by using operating system APIs (for instance see [3]) or Java Virtual Machines that allow to design graphics applications in Java, or by means of some ad-hoc software. Elite (`http://home.rochester.rr.com/ohommes/elite/`) is a fast 3D rendering engine for small devices running Java. This engine provides a framework to create and display 3D wireframe models. PocketGL (`http://pierrel5.free.fr/pocketglb/`) for Pocket PC (written in C and C++) allows to draw 3D objects and manage 3D transformations.

On the other hand, realistic visualization of large and complex models is not yet possible for the computational limitations of PDAs. To overcome this problem, solutions for hardware-accelerated remote rendering have been recently presented in [4] and [5]. These works aim to deploy hardware resources of centralized systems in order to allow the user to visualize and investigate large data set

models on PDAs. This kind of application is strictly related to the problem of transmitting video data streams to remote devices.

One of the first application of PDAs was for tourist and transportation purposes. Mobile devices can guide people through both real and virtual sites (museums, archeological sites, and so on). Two examples of virtual guides for museums and archeological site are presented in [6] and [7], respectively. On the other hand, PDAs were used to visualize maps of Virtual Harlem in [8]: maps allow to retrieve information about current location as well as moving to any place in Virtual Harlem. A mobile information system for public transportation is presented in [9]: this system helps travellers both to find the best public transport to reach the destination and to estimate the time needed to arrive.

Another field of application where the employment of mobile technologies seems to be very promising is telemedicine. Telemedicine scenarios include today in-hospital care management, remote teleconsulting, collaborative diagnosis, and emergency situations handling. Different types of information need to be accessed by means of heterogeneous client devices in different communication environments in order to enable high quality continuous sanitary assistance delivery wherever and whenever it is needed. Personal mobile telemedicine systems using wireless communication links have been employed in several applications and have been extensively studied [10, 11, 12]. Java, XML, and XSL technologies are used in [13] to ensure software portability and effective data presentation on heterogeneous access devices.

Education, entertainment, and training are fields where new technologies have always found large application. For instance, in [14], students of an elementary school can observe phenomena on a large display size and PDAs are used to aid data collection. Many commercial programs are available for entertainment on mobile devices; a near-exhaustive list of reviews and articles concerning games for PDAs can be found in `http://www.pdarcade.com/`.

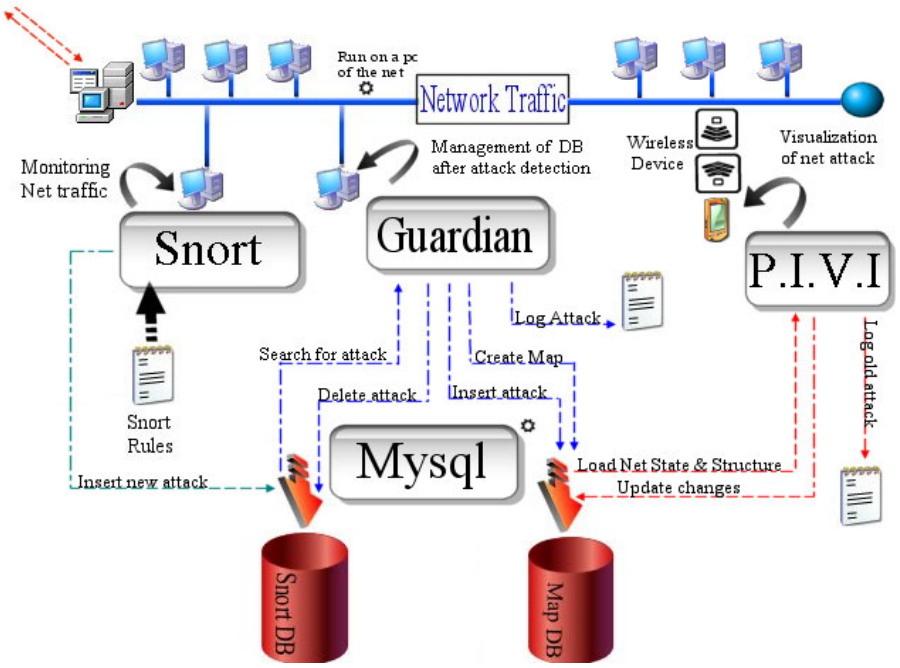

Figure 1: IMoViS architecture.

3 The proposed architecture

This section provides an overview of the proposed architecture. Details concerning components of the whole system are presented in sections 3.1, 3.2, and 3.3.

A complete scheme is shown in Figure 1. Three main components can be identified:

- Snort
- Guardian
- Portable Intrusion Visualization Interface (P.I.V.I)

Snort (`http://www.snort.org/`) IDS is able to monitor network traffic and uses a set of rules to identify attacks and intrusions (the terms attack and intrusion in this context refer to a violation of Snort's rules). Snort stores each attach into a database (Snort DB) shared with the second component of the system: Guardian.

Guardian is a program devoted to interface Snort to the visualization application and is not necessarily located on the same machine as Snort. Guardian manages the Snort DB in order to search for and delete attacks and produces a specific database (Map DB) used by the visualization application. Guardian can also produce log files useful for saving attack information and details.

P.I.V.I. is the visualization application. It loads a description of the building to be monitored as well as the network status information from Map DB. The description of the building can be organized at different levels of detail and contains a graphics representation of the site under analysis. It is possible to specify which data have to be monitored, such as IP address, user name, office telephone number, and so on. P.I.V.I. allows the user to delete already "processed" attacks from Map DB and to save log files of them.

MySQL (`http://www.mysql.com/`) have been chosen for the DBMS. MySQL has good performance in terms of speed and reliability and is open source. JDBC drivers allow the interaction between MySQL and the three system components: Snort, Guardian, and P.I.V.I.

Guardian and P.I.V.I. have been developed in Java and need a Java Virtual Machine (JVM) to be executed. Among the different Personal Java Application Environment implementations available for PDAs, the Insignia Jeode Runtime (`http://www.insignia.com/`) has been preferred to other implementations for its high adherence to Sun Microsystems PersonalJava 1.2 specifications and for its advanced performance in executing Java byte-code. Jeode Runtime can be used both as a Pocket Internet Explorer plug-in to run Java applets from a Web page and as a stand-alone JVM to run Java applications.

3.1 Description of Snort

Snort is a Network Intrusion Detection System well known among computer security professionals. Free and Open Source, it is rapidly becoming the tool of choice for Network Intrusion Detection. Unix and Windows versions are available and a huge and active enthousiast community of users contributes

to develop filters and rules (signatures) suited to discover intrusion attempts (and of course makes them free to other users). It is considered one of the most advanced intrusion detection system, free, but with the quality of a commercial product [15]. Snort is based on the sniffing libraries libpcap/winpcap [16, 19]. The detection engine uses detection rules written using a simple but powerful language that describes per packet tests and actions: logging, content pattern matching, and attacks and probes detection. Snort has real-time alerting capability: alerts are sent to syslog, via SMB messages, or written to a separate "alert" file. A database may also be used to store them. Snort is configured by using command line switches and Berkeley Packet Filter [17] commands. Third party add-ons may be used to simplify administration tasks.

3.2 Description of Guardian

Guardian has two main purposes:

1. interfacing Snort to P.I.V.I.;
2. providing an effective tool for map design.

Every time a rule violation is detected, Snort pushes a record into the database (Snort DB of Figure 1). Guardian manages data produced by Snort and generates a second database (Map DB of Figure 1) containing a sort of "meta-data" used for the visualization process. Only a sub-set of the information produced by Snort is required by P.I.V.I.

Guardian periodically polls Snort DB in order to detect new data insertions. Two cases may occur:

- a new attack or violation is detected;
- a new event is detected.

The first case causes a data insertion in Map DB in order to generate a new visual event; the second case concerns an already active attack and just the information regarding it is updated in Map DB.

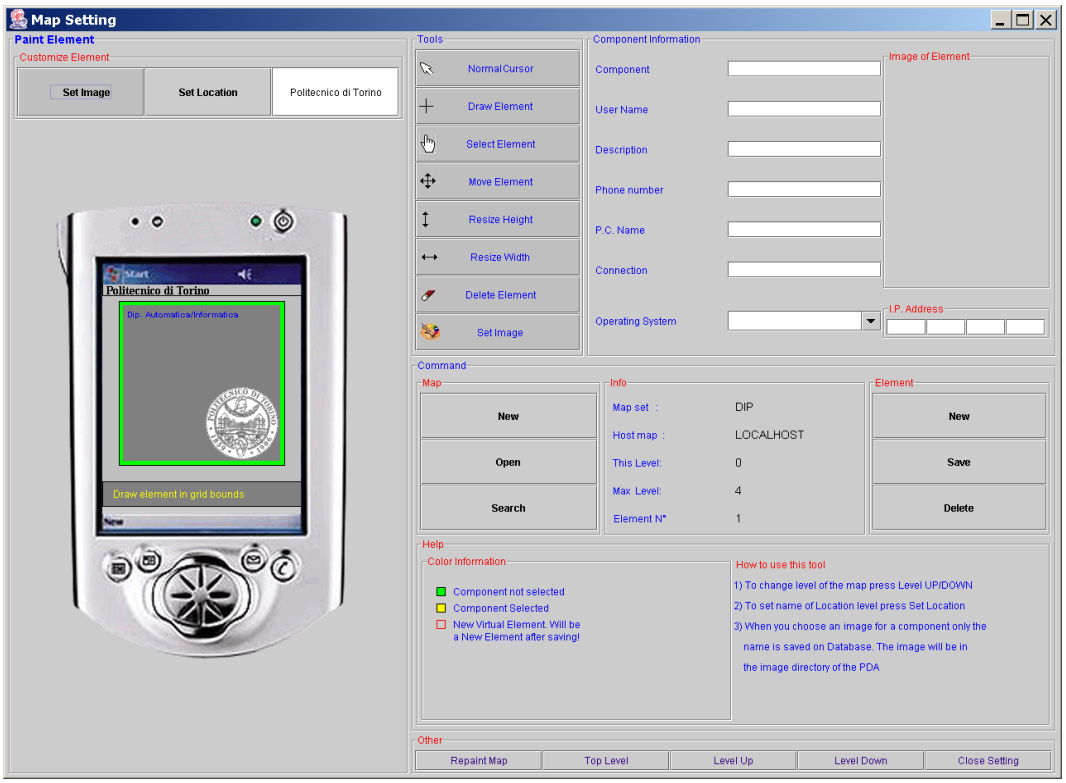

Figure 2: The interface for map design.

Snort inserts a new record each time a security rule is violated. After an attack has been processed, Guardian deletes the corresponding record in Snort DB keeping near constant the database size. The deletion of a set of records would lead to a loss of information; in order to avoid this, the user can configure Guardian to save deleted information into a log file (see Figure 1).

The tool for map design is entirely written in Java and it is shown is Figure 2. Three main zones can be identified: painting area (left part), information insertion area (upper-right part), and command area (lower-right part). The painting area allows to define rectangular areas over a grid. Rectangle sizes and spatial coordinates are intuitively drawn by the mouse. Each rectangle is an *element* that can be selected, moved, resized, and deleted. An image can be placed inside a rectangle and related information is introduced in the information area.

When a new map is designed, a new database is created using the *New* button of the command area. Existing databases can be also modified and deleted.
Building descriptions are organized in different levels. *Level up* and *Level down* buttons allow to move up and down through the levels hierarchy.

3.3 Description of P.I.V.I

This section will provide details of P.I.V.I. and, in particular, of the graphical interface. The visualization application has been developed and tested on a Personal Digital Assistant Device Compaq iPaq H3630 equipped with the Microsoft PocketPC operating system. Basic features of this PDA are: 206 MHz Intel StrongArm CPU, 4096 colors TFT LCD display, 240 $\times$ 320 pixels (2.26 $\times$ 3.02 inches) resolution touch screen, 32 MB RAM and 16 MB Flash ROM. Although the graphical interface has been tailored for PDA devices, P.I.V.I. can be used on every device equipped by a Java Virtual Machine (multi-channel interface).
The introductive screenshot of P.I.V.I. is shown in Figure 3. The main problem involved in designing graphical interfaces for mobile devices is the display size [20]. In particular large size images have to be accurately managed. Two solutions are possible:

- displaying an image larger than the display allowing the user to move and zoom it;
- representing the information by means of a set of hierarchical images.

The former approach is easier to implement and is recommended when the user must have a global view at the maximum level of detail; for instance, a radiograph has to be analyzed in its entirety. In these cases the user has to be able to move the image in order to "browse" it. This operation can require a lot of computational power and results to be particularly slow on low-end PDAs.
The latter strategy is suited for applications where hierarchical organizations of information may be obtained. The proposed intrusion detection system has been devised to monitor the network traffic

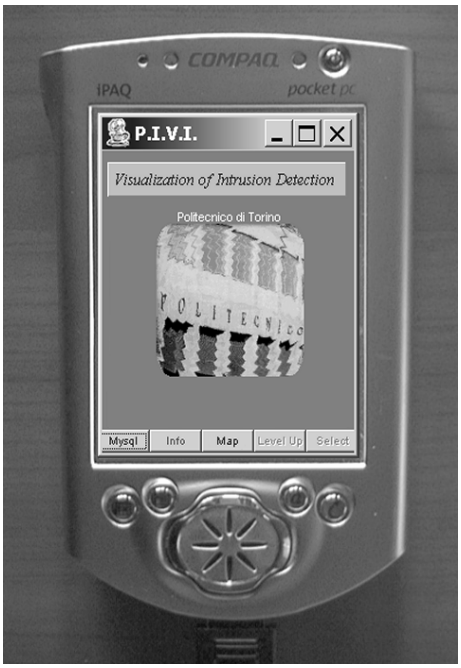

Figure 3: The introductive screenshot on the PDA.

within our Computer Science Department (DAUIN). The hierarchical representation of the building is shown in Figure 4. DAUIN is divided in three sectors; offices are organized as two rows (left and right) in the first and in the third sector, while, in the second sector, two corridors divide three rows of offices. Each office can contain an arbitrary (generally from one to four) machines.

The starting panel (see Figure 3) allows the user to configure a set of parameters to be used to connect to the database (Map DB of Figure 1). P.I.V.I. loads the map of a building from Map DB and polls it for new attacks. The colors used to draw each building contour are used to focus the attention of the user where a new attack has been detected.

An example is shown in Figure 5. The left image shows an attack to a machine placed in the department area labelled the *2nd sector* (the border of this sector is darker in order to denote an attack). More details may be found by browsing the lower level related to the red part. The lower level is shown in the right image of Figure 5; here the attack involves a machine placed in office N. 61. Selecting this office it is possible to retrieve all details of the attack.

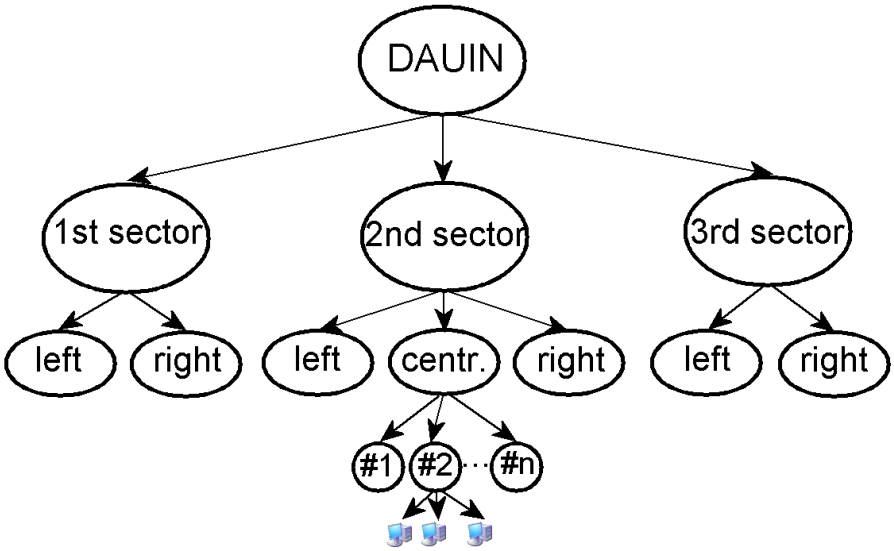

Figure 4: DAUIN hierarchical representation.

Figure 6 (left image) shows all information about the attack: IP address and operating system of the PC, user name and office telephone number, and date and hour of the attack. The user can get further information by pressing the Next button placed at the right lower corner (see Figure 6, right image). The kind of attack (in this case a `telnet bad login`) is shown together with the IP address and name of the remote machine performing the intrusion and, when available, the name of the user on the remote machine. A set of pages is automatically generated when the number of attacks affecting a machine grows.

An alarm is activated every time a machine monitored is attacked; this alarm is outlined by changing the color of the map from the lowest level of the hierarchy ("office level") to the highest level ("building level"). The user can deactivate the alarm by selecting a level by means of the Select button (see for instance Figure 5) and updating the status of the alarm. In this way, the alarm will be deactivated for the selected level and for all lower levels; for instance, a set of alarms concerning a unique part of the building can be deactivated at the same time. Moreover, the user can delete all information related to each attack from Map DB; in this case a log file contained deleted information can be generated.

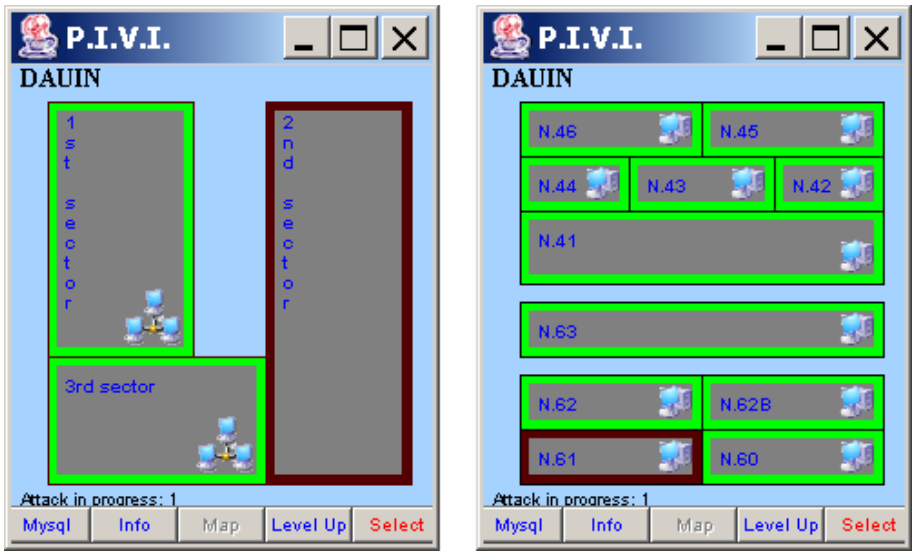

Figure 5: Two screenshots of an attack.

4 Performance Issues

In order to test the proposed system we developed a program able to generate "customizable" attacks over a LAN. The number of packets (every packet represents a violation to the Snort's rules) and the delay between packets are changed in order to measure the number of lost packets (a packet is denoted as lost when the corresponding rule violation has not been detected and processed). Two tests have been performed:

1. the system is stressed by transmitting a fixed number of packets (1000) and varying the delay between packets. This test aims to measure the number of packets correctly received and processed.

2. The system is stressed by transmitting a variable number of packets with a fixed delay (500ms). Also this test aims to measure the number of errors in the receiving and processing phases.

The results for the first test are reported in Figure 7. The graphic lists the number of lost packets (on the ordinate) for a delay varying from 200 ms to 0.001 ms. It can be noticed the number of lost

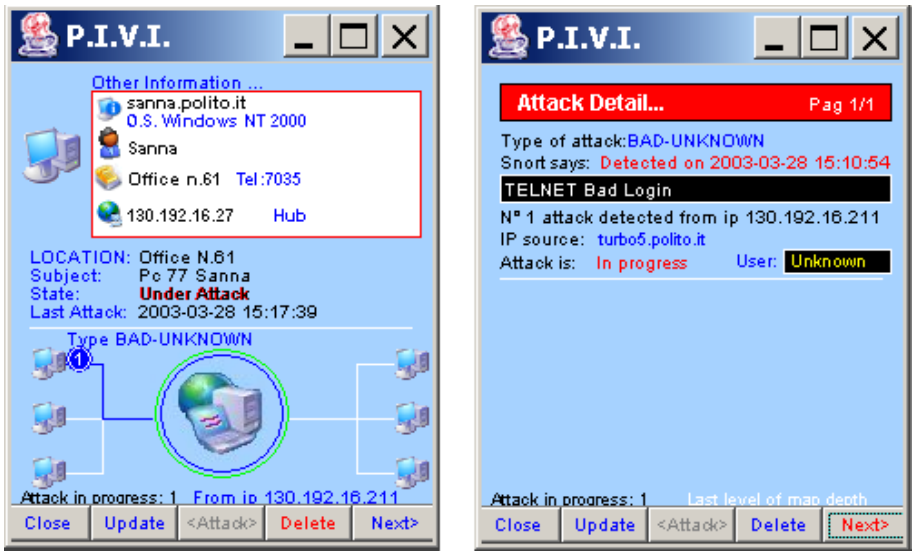

Figure 6: Details of a detected attack.

packets is almost independent of the delay; moreover, the percentage of error is 1.3% in the worst case.

Results of the second test are reported in Figure 8. The graphic lists the number of lost packets (on the ordinate) for a number of transmitted packets varying from 150 to 6000 ms. It can be noticed as the lost packets are negligible for a number of transmitted packets less than 1500, while the error is about 10% for 6000 packets transmitted.

5 Conclusions

Mobile devices, and in particular PDAs, are quickly changing the way information can be retrieved and visualized. Computational power and display size of a PDA are still two factors that limit the application of these new technologies when large data repositories have to be managed. On the other hand, Network Intrusion Detection Systems add a new level of visibility into the nature and characteristics of network traffic, identifying threats from unauthorized users, back-doors, and hackers. This paper proposes an architecture able to manage and visualize the large amount of data produced by

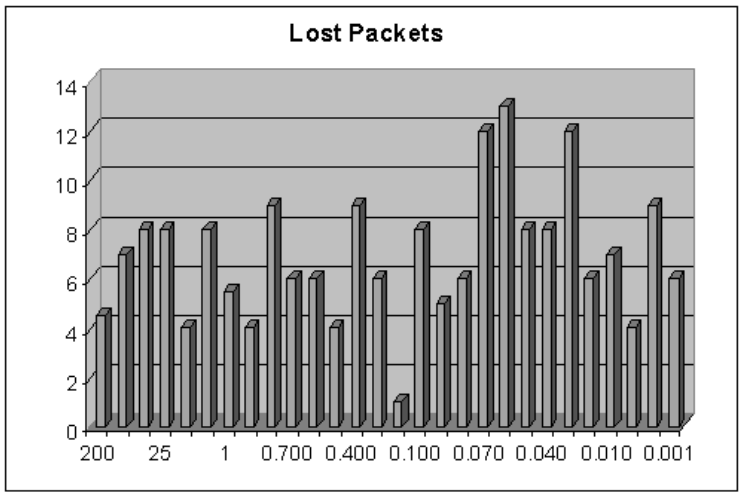

Figure 7: Test 1: lost packets.

means of an intrusion detection program. Every time an attack or an intrusion is detected an effective visual event in sent to the user's PDA. An ad-hoc tool allows the user to organize the building to be monitored by a set of different levels of detail. Machines are placed at the leaf level of the hierarchy and for each machine it is possible to specify the user, the IP address, the operating system, and so on. This hierarchical spatial data organization allows the user to efficiently and intuitively control large builds obtaining both a global view of the whole system as well as the detailed information concerning any incoming attack.

Acknowledgements

We want to thank Ing. Luca Lamorte for his support in implementing IMoViS.

References

[1] Card, S. K., Mackinlay, J. D., and Shneiderman, B. (1999) *Readings in Information Visualization Using Vision to Think*. Morgan Kaufmann Publishers.

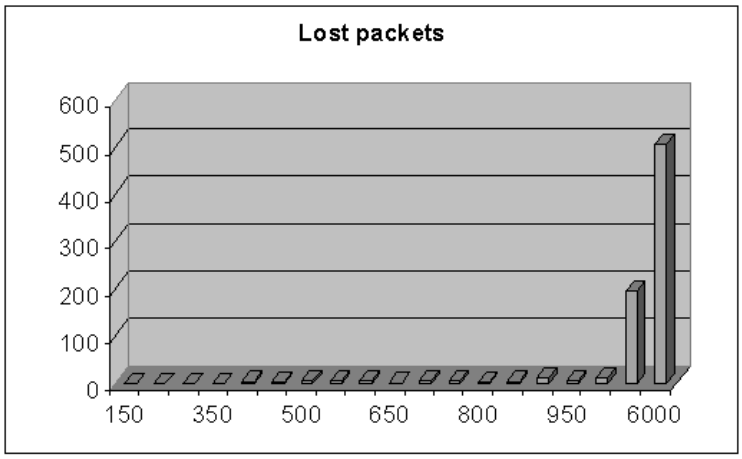

Figure 8: Test 2: lost packets.

[2] Want, R. and Borriello G. (2000) Survey on information appliances. *IEEE Computer Graphics & Applications*, **20:3**, 24-31.

[3] Fairuz Shiratuddin, M., Perdomo, J. L. and Thabet W. (2002). 3D Visualization Using the Pocket PC. *Proceedings of ECPPM 2002*, Portorož, Slovenia, 9-11 September.

[4] Stegmaier, S., Magallón, M. and Ertl, T. (2002). A generic solution for hardware-accelerated remote visualization. *Procceedings of EG/IEEE TCVG Symposium on Visualization VisSym02*, Barcelona, Spain, 27-29 May, pp. 87-94, ACM, New York.

[5] Lamberti, F., Zunino, C., Sanna, A., Fiume, A. and Maniezzo, M. (2003). An accelerated remote graphics architecture for PDAs. *Proceedings of Web3D 2003 Symposium*, Saint Malo, France, 9-12 March.

[6] Oppermann, R. and Specht M. (1998). Adaptive support for a mobile museum guide. *Proceedings of IMC'98*, Rostock, Germany, 24-25, November.

[7] Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P. and Almeida L. (2002). Archeoguide: an augmented reality guide for archeological

sites. *IEEE Computer Graphics & Applications*, **22:5**, 52-60.

[8] Johnson, A., Leigh, J., Carter, B., Sosnoski, J. and Jones. S. (2002). Virtual Harlem. *IEEE Computer Graphics & Applications*, **22:5**, 61-67.

[9] Preim, B., Fänger, A., Goetze, M., Rainer, M. (1999). Guiding travellers by a mobile information system for public transportation. *Proceedings of the Workshop on Adaptive Design of Interactive Multimedia Presentations for Mobile Users*, Sitges, Spain, 7 March.

[10] Istepanian, R. S. H. (1998). Modeling of a GSMbased mobile telemedical system. *Proceedings of the 20th Conference of the IEEE Engineering in Medical and Biology*, Hong Kong, 29 October - 1 November, pp. 926-930.

[11] Istepanian, R. S. H. and Petrosian, A. A. (2000). Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system. *IEEE Transaction On Information Technology in Biomedicine*, **4:3**, 189-194.

[12] Woodward, B., Istepanian, R. S. H. and Richards, C. I. (2001). Design of a telemedicine system using a mobile telephone . *IEEE Transaction on Information Technology in Biomedicine*, **5:1**, 13-15.

[13] Lamberti, F., Montrucchio, B., Sanna, A. and Zunino C. (2002). A Web-based architecture enabling multichannel telemedicine applications. *Proceedings of SCI'02*, Orlando, Florida, 14-18 July, **XIII**, pp. 257-262.

[14] Johnson, A., Moher, T., Cho Y. J., Lin Y. J. Haas, D. and Kim J. (2002). Augmenting Elementary School Education with VR. *IEEE Computer Graphics & Applications*, **22:2**, 6-9.

[15] Northcutt, S., Novak, J., McLachlan, D., (2001) *Network Intrusion Detection - An Analyst's Handbook (2nd Edition)*. New Riders.

[16] The Tcpdump Group: *pcap, Packet Capture Library.* 1988-1994-2000, Original authors: Van Jacobson, Craig Leres and Steven McCanne, all of the Lawrence Berkeley Laboratory, University of California, Berkeley, CA.

[17] McCanne, S., Jacobson, V., The BSD packet filter: A new architecture for user-level packet capture". In USENIX Technical Conference Proceedings, pages 259- 269, San Diego, CA, Jan. 1993.

[18] The Tcpdump Group: *TCPDUMP 3.5, a tool for network monitoring and data acquisition.* 1988-2000, Original authors: Van Jacobson, Craig Leres and Steven McCanne, all of the Lawrence Berkeley Laboratory, University of California, Berkeley, CA.

[19] Risso, F. and Degioanni, L. (2001) An Architecture for High Performance Network Analysis. *Proceedings of the 6th IEEE Symposium on Computers and Communications*, Hammamet, Tunisia, 686-693.

[20] Kwang B. L., Roger G. and Watt, J. (2003). Zoomable User Interfaces for Personal Digital Assistants. *to be published in the Journal of IEEE Transactions on Professional Communication.*