International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 1
ISSN: 2454-4248
411 – 414

Implementation of Lessar Algorithm for Analysis of Network Intrusion Detection System in Wireless Sensor Networks

Suman, D/O Raghubir Singh
JRF Qualified

ABSTRACT: Wireless sensor networks are the result of multiple technological advances in electronics, nanotechnology, wireless communications, computing power, network development and robotics. Compose distributed systems devices usually composed of integrated, including at least CPU, radio, and sensors / actuators number. The wireless sensor networks (WSN) heterogeneous networks are formed by sensors, gateways and backend resources very limited physical.

The sensors can measure parameters such as temperature, movement, lighting, humidity, etc..; the gateways establish the link with networks traditional and familiar. The back ends are responsible for the processing and display unit the captured data. Although several studies showed WSN middleware, has not been achieved with this industry acceptance due mainly to the different methodologies programming. The teams consist of WSN lowest consumption, costs and form factors. The reality is quite different environments applications are supported with equipment more powerful and fed by redundant power networks. The current article describes then implementation of lesser algorithm for analysis of network intrusion detection system in wireless sensor networks.

KEYWORDS: *Wireless, Network, Sensor*

*****

I. INTRODUCTION

The work aims to define a way that developers can use technologies existing compatible standard to describe processes and services offered by WSN, without extra coding. The advantages of the method are Two: immediate adaptation of a WSN company, and integration of services WSN applications for higher-level simple modifications. The reality is that the main problems encountered in programming WSN relate the different based programming, as well with heterogeneity in both hardware and operating systems. While businesses and offices in the applications are supported with high-quality equipment and computing power are powered by redundant power networks; WSN are optimized for scenarios minimum energy consumption, lower costs and reduced form factors.

Since the early 90s, has sought to achieve this goal with modeling, analysis and adaptation processes business. So architectures have appeared service-oriented, which are based on Internet (SOA (2)). In parallel, the WSN have achieved such a degree of development that is considered as an integral part of the Internet the future, achieving extend the domain of this network to the real world. The two trends, together form the basis of a new type of applications where devices interact processes in an impossible way imagine a few years ago. This was true for single sensor nodes up to application servers'scale, with this, and if the trend is not declined, the captured by the WSN data would influence the information flow of current processes real time, and could even trigger new processes.

To achieve this level of interaction, WSN should relate unfailingly existing SOA with current technologies, such as XML (3), Web Services (4), and Execution Language Business Process (BPEL (5)), to name only a few. However, due to the high demand for resources of these applications are difficult applicable in restricted environments of WSN.

Some common features of WSN applications are defined by the own taxonomy of these networks. Not a good idea to add support for each of the platforms in the tools development. This is rationalized by the addition of an integration layer between the hardware and application, known as middleware. The middleware has been the subject of research Distributed systems for many years. Create WSN middleware for a challenge, considering the restrictions set by this technology, methods well known as CORBA or Enterprise Java Beans should be discarded by excessive requirements computing power and memory. Furthermore, by the instability of communications in the WSN environment, methods traditional client-server is not recommended.

II. RESEARCH STUDY

In the LESSAR algorithm, a global time is maintained in wireless sensor networks by organizing the whole network system into levels. Level discovery is performed initially when the network is deployed. Sink which collects information from all nodes forms the root and is assigned level 0. It broadcasts level discovery packet to its neighbors. Nodes receiving the packets are assigned level 1 and broadcast the level discovery packet to the other nodes.

One node may as a result, receive many packets but it accepts only the one with the lowest level as its ancestor and takes its value +1 as its own level. Thus broadcasting continues. All the sensor nodes are connected in this hierarchical network topology. When a new node enters, it

 ISSN: 2454-4248
411 – 414

broadcasts the level request packet to enquire the current level values of its neighbors. From the responses obtained, it selects the smallest one + 1 as its level. On node failure, its children notice this, when its timer of observing keep alive message expires. These nodes broadcast level request packet and redo the level discovery process again.

In LESSAR, nodes are synchronized level by level. Each node believes that the clocks in its upper level are accurate than its local clock and synchronize with them. It only accepts time sync packets from the upper level and drops all others from the lower levels. So the whole wireless sensor network follows the clock of the sink. This will be synchronized by GPS/NTP. This method has very low resource consumption and computation complexity.

To deal with the energy management problem, different power management schemes are discussed here. The most important constraint in all wireless sensor networks is the Energy efficiency problem since they are equipped with limited power sources. So an efficient power management should be adopted. Research is conducted using static approaches to attain power management by making the nodes which are not currently being utilized to go to low power states but this should be decided earlier, in a fixed time schedule and not at run time.

Dynamic Power Management (DPM) is widely used in wireless sensor networks. During run time, dynamic techniques can further improve the reduction in power consumption by selectively shutting down the hardware components. After designing a system, additional power savings can be obtained by Dynamic Power Management. Protocols and algorithms have to be tuned for an application. Embedded operating systems and software become a critical requirement of such networks.

Major consumer of energy in a wireless sensor network is the energy communication circuits. So communication should be performed only when needed. DPM should always consider when a node should go to sleep/idle state and how long it should remain there. Sensor nodes communicate using short data packets which have more dominance of startup energy. External events represent the interaction between the sensor node and the environment.

So DPM involves shutting down the sensor node during no event and waking them up when needed. So good energy saving is achieved. But sensors communicate using short data packets. So there is more dominance of startup energy. Therefore DPM should be carefully implemented. Operation in energy saving mode becomes energy efficient only if the time spent in that mode is greater than a decided Threshold. The common DPM policies are the Predictive policy and the Stochastic policy.

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or security standard practices. To understand the meaning of intrusion detection, we can use an analogy to the common “burglar alarm”. Just like the burglar alarm, intrusion detection works on a computer system or network and is enabled to detect possible violations of security policies and raise an alarm to notify the proper authority.

III. DISCUSSION

A Network Intrusion detection System (NIDS) is an intrusion detection system that tries to detect malicious activity such as service attacks, port scans or even attempts to break into computers by monitoring network traffic.

It is a fact that most firewalls are configured and deployed by humans. And human beings are prone to error making. This knowledge is well-known to the intruders who try to take advantage of it. They try to find a security breach in the configuration of the firewall and exploit it.

As most organizations deploying security mechanisms use encryption for protecting files and external network connections, so the intruder’s interest will lie on such locations where the encryption/protection of data transmission is missing or very minimal. This is generally the case where the data is stored and/or transmitted to trusted hosts and networks. Even if a VPN request connection is made between trusted hosts and networks and the main network in question, attacks by intruders can be very efficient. Furthermore, the probability of successful attacks in this type of attacks can be very high as, in most cases, not even minimum encryption is used for increasing performance.

Address spoofing is a method which is used to hide the real address of the sender of a network packet, particularly the intruder. However, this can also be used to bypass the firewall and gain unauthorized access to a network or computer. Contemporary firewalls have in-built mechanisms to avoid this fraud. They are, practically, not deceived by this kind of address spoofing. But the principle of address substitution, in itself, remains an urgent issue that needs to be addressed.

For instance, an intruder can mask her own address with the address of a trusted network address or with the address of a trusted host in the network and send packets containing malicious data which may adversely affect the network computers and data. This method is different from the attack by trusted host and network problem since in this case only the address of the trusted host is used rather than masquerading as the trusted host in the former case.

As the firewall software and hardware are built by humans, they themselves are prone to attack from the intruders. A successful attack on the firewall can lead to very serious consequences as once successfully attacked, intruders can

 ISSN: 2454-4248
411 – 414

freely access the resources of the protected network without the risk of being detected and traced. Moreover, an intruder can also tweak with the configuration and rules of the firewall to allow other kind of intruders to attack the network.

A network intrusion detection system reads all incoming packets and tries to find suspicious patterns known as signatures or rules. These rules are decided by a network administrator while the configuration and deployment of the network intrusion detection system based on the security and network policies of the organization. For instance, if it is observed that a particular TCP connection requests connection to a large number of ports, then it can be assumed that there is someone who is trying to conduct a port scan of all/most of the computers of the network.

A network intrusion detection system is not limited to inspecting the incoming network traffic only. Patterns and outgoing intrusion can also be found from the outgoing or local traffic as well. Some attacks might also come from the inside of the monitored network, as in trusted host attack.

At the heart of every modern network intrusion detection system there is a string matching algorithm. The network intrusion detection system uses the string matching algorithm to compare the payload of the network packet and/or flow against the pattern entries of the intrusion detection rules, which are a part of every network having a network intrusion detection system.

In many cases, intruders try to and penetrate firewalls to gain unauthorized access to corporate networks. This is done by attacking the firewall itself and breaking it down by tweaking its rules and signatures. In this case, the network intrusion detection system can decrease the risk of such attacks by temporarily backing up firewalls. The network intrusion detection system of this type filters packets based on their IP packet header. This enables the network administrator to deploy network intrusion detection systems with functionality comparable to that of very advanced firewalls. Further, this type of network intrusion detection system can also be used while the general firewall is down for maintenance or when the firewall software is being updated or for any other reason.

IV. SIGNIFICANCE OF THE STUDY

Generally functions of controlling file access are done to specialized systems, such as Secret Net, which are intended specifically for protecting network information from unauthorized access. However, protection of some critically important files such as database files and password files cannot be done by such systems. Moreover, such systems are mainly developed for the Windows and NetWare platforms. So such systems fail in UNIX environments which are used for network applications in many organizations. So in such types of cases a network intrusion detection system comes to the rescue of network administrators. Mainly host based network intrusion detection systems are used in such cases which are based both on log-file analysis (Real Secure Server Sensor) and IDSs analyzing system calls (Cisco IDS Host Server).

A network intrusion detection system can help in identifying the address of unknown/external hosts within the protected network segments. It can also detect increased traffic and special kind of activities from specific workstations which were not involved in such kind of activities before. Such activities can be a hint to malicious activities from the hosts and the network administrator must be informed about this.

Firewalls are essential for protecting the corporate network from unwanted network activities. But a firewall can work desirably only when it is configured correctly. Incorrect configuration and inefficient testing of a firewall can wreck havoc on the network. Installing a network intrusion detection system before and after the firewall allows one to test the efficiency of the firewall by comparing the number of attacks before and after the firewall. In addition to this, it can also act as a backup for the firewall.

V. CONCLUSION

Log files from routers and other network equipment can serve as an additional source of information on the various attacks that a data network can be prone to. However, most organizations do not analyze this collected information because it is a time overhead for the organization and the tools available for such analyses (such as net Forensics) are rather costly.

A network intrusion detection system can be configured to do this work. The task of collecting such log-file information and analyzing logged security events can be delegated to the intrusion detection system, which in this case, serves as a Syslog server. It can centralize such tasks of collecting log-file information and detect attacks and misuse of the network. It also prevents unauthorized modifications of the events logged. Moreover, the events logged are immediately sent to another server so that the intruder can't remove any traces after completing her operation.

Most network administrators use the default network configurations for simplifying their tasks. But this also simplifies the task of an intruder because she knows the default network configurations. This makes the network more vulnerable and open to successful attacks. A network intrusion detection system can be configured to search the hosts where default configurations have been used and can also recommend corrective measures that can be taken.

REFERENCES

[1] Gungor, V. C., Lu, B., & Hancke, G. P. (2010). Opportunities and challenges of wireless sensor networks

ISSN: 2454-4248
411 – 414

in smart grid. *Industrial Electronics, IEEE Transactions on*, *57*(10), 3557-3564.

[2] Ergen, S. C., & Varaiya, P. (2010). TDMA scheduling algorithms for wireless sensor networks. *Wireless Networks*, *16*(4), 985-997.

[3] Akyildiz, I. F., & Vuran, M. C. (2010). *Wireless sensor networks* (Vol. 4). John Wiley & Sons.

[4] Stabellini, L., & Zander, J. (2010). Energy-efficient detection of intermittent interference in wireless sensor networks. *International Journal of Sensor Networks*, *8*(1), 27-40.

[5] Chiwewe, T. M., & Hancke, G. P. (2012). A distributed topology control technique for low interference and energy efficiency in wireless sensor networks. *Industrial Informatics, IEEE Transactions on*, *8*(1), 11-19.

[6] Alemdar, H., & Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, *54*(15), 2688-2710.

[7] Baccour, N., Koubaa, A., Mottola, L., Zuniga, M. A., Youssef, H., Boano, C. A., & Alves, M. (2012). Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, *8*(4), 34.

[8] Tang, L., Sun, Y., Gurewitz, O., & Johnson, D. B. (2011, May). EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (p. 23). ACM.

[9] Incel, O. D., Ghosh, A., Krishnamachari, B., & Chintalapudi, K. (2012). Fast data collection in tree-based wireless sensor networks. *Mobile Computing, IEEE Transactions on*, *11*(1), 86-99.

[10] Baccour, N., Koubaa, A., Mottola, L., Zuniga, M. A., Youssef, H., Boano, C. A., & Alves, M. (2012). Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, *8*(4), 34.

[11] Tang, L., Sun, Y., Gurewitz, O., & Johnson, D. B. (2011, May). EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (p. 23). ACM.

[12] Incel, O. D., Ghosh, A., Krishnamachari, B., & Chintalapudi, K. (2012). Fast data collection in tree-based wireless sensor networks. *Mobile Computing, IEEE Transactions on*, *11*(1), 86-99.