ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

INTRUSION TOLERANCE MULTIPATH ROUTING IN HETEROGENEOUS WIRELESS SENSOR NETWORK

Fazeel Irshad Zama, Suchit R. Gajbhiye
[*] Department of Computer Engineering
Wainganga College of Engineering NAGPUR
Dist NAGPUR(M.S) INDIA

ABSTRACT

Trust based approaches have been widely used to counter insider attacks in wireless sensor networks because traditional cryptography-based security mechanisms such as authentication and authorization are not effective against such attacks. A trust model, which is the core component of a trust mechanism, provides a quantitative way to evaluate the trustworthiness of sensor nodes. The trust evaluation is normally conducted by watchdog nodes, which monitor and collect other sensors' behavior information.

Most existing works mainly focus on the design of the trust models and how these models can be used to defend against certain insider attacks. However, these studies are empirical with the implicit assumption that the trust models are secure and reliable. In this paper, we discuss several security vulnerabilities that watchdog and trust mechanisms have, examine how inside attackers can exploit these security holes, and finally propose defending approaches that can mitigate the weaknesses of trust mechanism. We observe that many existing trust models adopting watchdog as their monitoring mechanism do not explicitly address these weaknesses. Our goal in this paper is to demonstrate how serious insider attacks can be.

KEYWORDS: Network security, virtual network system computing, intrusion detection, attack graph, zombie detection

INTRODUCTION

Insider threat is an important security issue in wireless sensor network (WSN) because traditional security mechanisms, such as authentication and authorization,cannot catch inside attackers who are legal members of the network. Inside attackers can disrupt the network by dropping, modifying, or misrouting data packets. This is a serious threat for many applications such as military surveillance system that monitors the battlefield and other critical infrastructures.Trust mechanism with the notion of trust in human society has been developed to defend against insider attacks . Since WSNs consist of hundreds or thousands of tiny sensor nodes, the trust mechanism is often implemented as a distributed system where each sensor can evaluate, update, and store the trustworthiness of other nodes based on the trust model.

Thus, an inside attacker can disguise its malicious behavior behind network traffic or noise. Third, we cannot ignore the fact that insiders have internal knowledge about our network and security mechanisms against attacks. By exploiting such knowledge, inside attackers can launch their attacks intelligently to avoid being detected. We observe that many existing trust models adopting as their monitoring mechanism do not explicitly address these weaknesses. Our goal demonstrate how serious insider attacks can be in WSNs even with the presence of trust mechanism, and to introduce defending approaches to improve the trust mechanism.

PROBLEM DEFINITION

The challenge is to establish an effective vulnerability/attack detection and response system for accurately identifying attacks and minimizing the impact of security breach to virtual network system users. In a virtual network system where the infrastructure is shared by potentially millions of users, abuse and nefarious use of the shared infrastructure benefits attackers to exploit vulnerabilities of the virtual network system and use its resource to

deploy attacks in more efficient ways. Such attacks are more effective in the virtual network system environment since virtual network system users usually share computing resources.

METHODOLOGY

Least-Disruptive topology Repair (LeDiR) algorithm LeDiR introduces significantly less messaging overhead to enable and during the recovery in comparison to the centralized version and RIM,. Actually, in the centralized version, each node must be aware of the complete network topology, which involves N^2 messages required for maintaining the network status, as pointed out earlier. Thus, the messaging overhead dramatically grows as the node count increases. On the other hand, RIM requires maintaining one-hop neighbor information for performing the recovery. Thus, an extra N message overhead is considered for RIM to exchange information initially at the network startup. Conversely, LeDiR leverages the available route discovery process and does not impose prefailure messaging overhead. The only communication cost incurred during the recovery is when a node informs its children about its movement or broadcasts the successful relocation.Nonetheless, as previously noted, the avoidance of explicit state update comes at the cost of increased travel Overhead. It is important to note that for the results no heartbeat messages are counted during the network operation for all approaches. In practice, heartbeat messages may or may not be explicitly transmitted. Typically, a node that stays quiet for a long time has to send a message to confirm its healthy status. Otherwise, messages that are part of the normal network operation, such as route update, data packet generation, inter-the coordination, etc., would suffice. We argue that the number of heartbeat messages would vary from node to node and over time. It is our view that they are not part of the recovery process in case a node failure is to be tolerated. Therefore, we did not fthe in heartbeat messages in the results of LeDiR, RIM, and the centralized approach. Path Length Validation Metrics: LeDiR does not extend the shortest path between any pair of nodes. We compare LeDiR to RIM and DARA. As expected, LeDiR achieves its design objective and does not extend any shortest path unlike RIM and DARA. RIM engages allneighbors of the failed node and triggers subsequent cascaded relocation. This can be tolerated in sparse topologies. However, in highly connected networks, i.e., large N or r values, many nodes are involved in the recovery process, as indicated by . As a result, the scope of node movement grows dramatically, and the number of extended paths increases, as On the other hand, DARA performs very close to LeDiR in highly connected topologies. In sparse networks, DARA does not do well with significant number of extended paths.

Fig1. Trust Path generation

Fig 2. Path Regeneration

Trust mechanism

In general, trust mechanism works in the following stages. Node behavior monitoring : Each sensor node monitors and records its neighbors' behaviors such as packet forwarding. This collected data will be used for trustworthiness evaluation in the next stage. Watchdog is a monitoring mechanism popularly used in this stage. The confidence of the trustworthiness evaluation depends on how much data a sensor collects.

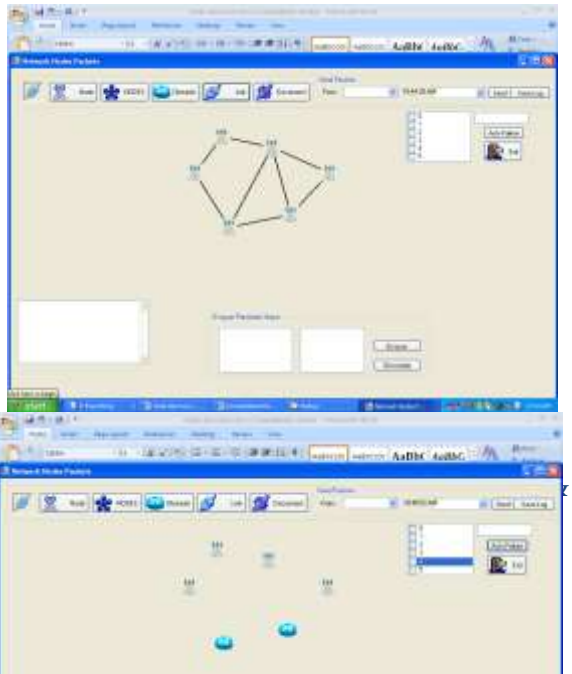
Trust measurement

Trust model defines how to measure the trustworthiness of a sensor node. introduced several representative approaches to build the trust model, which include Bayesian approach, Entropy approach, Game-theoretic approach, and Fuzzy approach. The trust value of a node may be different when we use different trust models. For example, when a node is observed to forward the packet n times and drops the packet Insider trust Management Intelligent inside attacks against trust mechanism Vulnerabilities in the inside attacker detection stage Average End-to-End delay Packet Delivery Ratio Energy Consumption Multi-hop Chain Topology Inside attack detection : Based on the trust value, a sensor node determines whether its neighbor is trustworthy for collaboration (such as packet forwarding). If a neighbor's trust value is less than a certain threshold , it will be considered as an untrusted or malicious node. Depending on the WSN's trust mechanism, the detection of such insider attacker may or may not be broadcast to the rest of the nodes in the SN.Moreover, we cannot keep aside the case of zero day attack where the vulnerability is discovered by the attacker but is not detected by vulnerability scanner. In such case, the alert being real will be regarded as false, given that there does not exist corresponding node

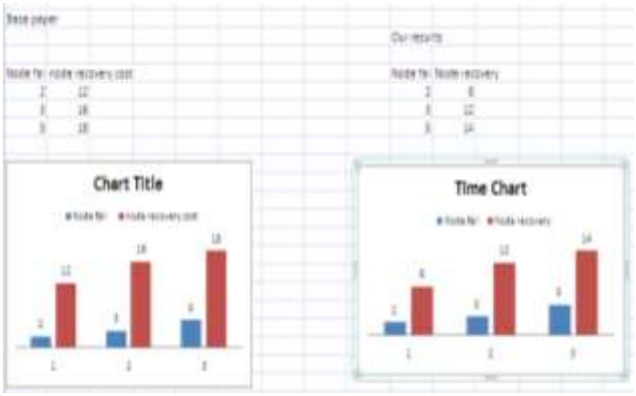
RESULT ANALYSYS

Failure detection: Thes will periodically send heartbeat messages to their neighbors to ensure that they are functional, and also report changes to the one-hop neighbors. Missing heartbeat messages can be used to detect the failure of thes. Once a failure is detected in the neighborhood, the one-hop neighbors of the failed the would determine the impact, i.e., whether the failed node is critical to network connectivity. This can be done using the SRT by executing the well-known depth-first search algorithm.

1. Smallest block identification: LeDiR limits the relocation to nodes in the smallest disjoint block to reduce the recovery overhead. The smallest block is the one with the least number of nodes and would be identified by finding the reachable set of nodes for every direct neighbor of the ailed node and then picking the set with the fewest nodes.
   Since a critical node will be on the shortest path of two nodes in separate blocks, the set of reachable nodes can be identified through the use of the SRT after excluding the failed node. In other words, two nodes will be connected only if they are in the same block.
2. Replacing faulty node: If node J is the neighbor of the failed node that belongs to the smallest block, J is considered the BC to replace the faulty node. Since node J is considered the gateway node of the block to the failed critical node (and the rest of the network), we refer to it as "parent." A node is a "child" if it is two hops away from the failed node, "grandchild" if three hops away from the failed node, and so on..Distributed LeDiR Implementation.
3. The foregoing discussion has assumed that nodes are aware of the network topology and can assess the impact of the failure and uniquely identify which node should replace the failed the. If every node in the network is communicating with all the other nodes, it would be possible to fully populate the routing table and for the individual nodes to reach consistent decisions without centralized coordination.



ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785



CONCLUSION

A trust threshold can be designed in static manner or dynamic manner. Static trust threshold might be optimal only for limited cases that we consider in the simulation. As a result, it may not be good for unconsidered situations. Meanwhile, dynamic trust threshold that adaptively changes according to situations in our network may have reasonably good results, although it may not be optimal for all situations. However, since dynamic trust threshold will be frequently computed, it must be designed in an energy-efficient way. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

REFERENCES

1. Azahdeh Faridi et al, "Comprehensive Evaluation of the IEEE 802.15.4 MAC Layer Performance With Retransmissions," IEEE Transactions on Vehicular Technology, Vol.59, No.8, October 2010, pp. 3917-3932.
2. Tran Hoang Hai and Eui-Nam Huh, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks Using Two-hops Neighbor Knowledge," Seventh International Symposium on Network Computing and Applications (NCA '08), July 2008, pp. 325-331.
3. K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of Attack and Defense Techniques for Reputation Systems," ACM Computing Surveys, Vol 41, Issue 4, 2009.
4. H. Takabi, J.B. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Dec. 2010.
5. "Open vSwitch Project," http://openvswitch.org, May 2012.
6. Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J Barker, "Detecting Spam Zombies by Monitoring Outgoing Messages," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 198-210, Apr. 2012.
7. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee "BotHunter: Detecting Malware Infection through IDS-driven Dialog Correlation," Proc. 16th USENIX Security Symp. (SS '07), pp. 12:1-12:16, Aug. 2007.
8. G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," Proc. 15th Ann. Network and Distributed Sytem Security Symp. (NDSS '08), Feb. 2008.
9. O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing, "Automated Generation and Analysis of Attack Graphs," Proc. IEEE Symp. Security and Privacy, pp. 273-284, 2002,
10. "NuSMV: A New Symbolic Model Checker," http://afrodite.itc. it:1024/nusmv. Aug. 2012.
11. P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graphbased network vulnerability analysis," Proc. 9th ACM Conf. Computer and Comm. Security (CCS '02), pp. 217-224, 2002.
12. X. Ou, S. Govindavajhala, and A.W. Appel, "MulVAL: A Logic- Based Network Security Analyzer," Proc. 14th USENIX Security Symp., pp. 113-128, 2005
13. R. Sadoddin and A. Ghorbani, "Alert Correlation Survey: Framework and Techniques," Proc. ACM Int'l Conf. Privacy, Security and Trust: Bridge the Gap between PST Technologies and Business Services (PST '06), pp. 37:1-37:10, 2006.
14. L. Wang, A. Liu, and S. Jajodia, "Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts," Computer Comm., vol. 29, no. 15, pp. 2917-2933, Sept. 2006.
15. S. Roschke, F. Cheng, and C. Meinel, "A New Alert Correlation Algorithm Based on Attack Graph," Proc. Fourth Int'l Conf. Computational Intelligence in Security for Information Systems, pp. 58-67, 2011.
16. A. Roy, D.S. Kim, and K. Trivedi, "Scalable Optimal Countermeasure Selection Using Implicit Enumeration on Attack Countermeasure Trees," Proc. IEEE Int'l Conf. Dependable Systems Networks (DSN '12), June 2012.
17. N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 1, pp. 61-74, Feb. 2012.
18. Open Networking Fundation, "Software-Defined Networking: The New Norm for Networks," ONF White Paper, Apr. 2012.
19. "Openflow," http://www.openflow.org/wp/learnmore/, 2012.
20. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow Enabling Innovation in Campus Networks," SIGCOMM Computer Comm. Rev., vol. 38, no. 2, pp. 69-74, Mar. 2008