Effectiveness of Trust Reasoning for Attack Detection in OLSR

Asmaa Adnane[1], Christophe Bidan[1] and Rafael T. de Sousa Jr[2]

[1] SUPELEC, SSIR team (EA 4039), 5 Av de la Boulaie, 35510-Cesson-Sévigné, France

[2] University of Brasília - SSIR team (EA 4039)
Av L3 Norte - FT - ENE, 70910-900 Brasília, Brazil

Abstract. Previous works [2, 3, 6] have proposed to check information consistency and to detect misbehavior nodes for the OLSR protocol based on semantic and trust properties. The basic idea is that each node uses only local observations to detect attacks without having to collaborate with other nodes. The objective of this paper is to prove the effectiveness of such approaches by presenting simulation results.

1 Introduction

Several studies have been carried on to secure protocols for ad-hoc networks, where nodes communicate directly with each other to relay messages without the support of a central entity. We are interested by the Optimized Link State Routing protocol (OLSR) [5]. This protocol presents an optimization of the classical link state algorithm adapted to the requirements of a mobile wireless LAN. The concept used in the protocol is the multipoint relays (MPRs). MPRs are nodes which broadcast messages during the flooding process. This method substantially reduces the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message.

In ad-hoc networks, the establishment of the routing table is associated to a process of trust construction through cooperation among nodes for discovering neighbors, selecting routers and announcing topology information. Thus, each node has to verify the expected scheduling and content consistency of protocol messages, enabling the mistrust of the other misbehavior nodes during this process [2, 3].

In previous works [2, 4] we have proposed for OLSR the integration of semantics checking and trust reasonings into each node, so as to allow a self-organized control to help nodes to detect attacks about modification of OLSR control messages. In addition, our proposal does not change the OLSR protocol and is compatible with the bare OLSR. The objective of this paper is to prove the effectiveness of our approach to the integration of trust reasoning in OLSR protocol to detect attacks by presenting the simulation results.

The paper is organized as follows. Section 2 presents notations and the trust specification language. Section 3 summarizes our previous works. Synthesis and simulation of

Adnane A., Bidan C. and T. de Sousa Jr R. (2008).
Effectiveness of Trust Reasoning for Attack Detection in OLSR.
In *Proceedings of the 6th International Workshop on Security in Information Systems*, pages 151-157
DOI: 10.5220/0001737501510157

our approach are presented in section 4. Finally, we conclude in Section 5, and present our future works.

2 Notations

In OLSR, each node maintains its local vision of the network. This vision consists in the following sets:

- $MANET$: the set of the whole MANET nodes,
- NS_X (Neighbor Set): the set of symmetric neighbors of the node X,
- $2HNS_X$ (2-Hop Neighbor Set): the set of 2-hop neighbors of the node X,
- $MPRS_X$: the set of nodes selected as MPR by the node X ($MPR_X \subseteq NS_X$), that is the nodes that are in charge of routing and forwarding the packets sent by X.
- $MPRSS_X$ (MPR Selection Set): the set of symmetric neighbors which have selected the node X as MPR ($MPRSS_X \subseteq NS_X$),
- RT_X (Routing Table): the routing table of the node X,

The node collects the information needed to maintain its local vision of the network by exchanging HELLO and TC messages. For these messages, we note:

- $X \overset{HELLO}{\longleftarrow} Y$, $X \overset{TC_Y}{\longleftarrow} Y$: respectively, the reception by node X of HELLO and TC messages from node Y ($HELLO = LS_Y$ and $TC_Y = MPRSS_Y$),
- $X \overset{TC_X}{\longrightarrow} *$, $X \overset{DATA_X}{\longrightarrow} *$: The broadcast by X of a TC or respectively a data message to be forwarded by its MPRs.
- $X \overset{TC_Y}{\nleftarrow} Y$: absence of an awaited TC message from node Y,
- $X \overset{DATA_X}{\nleftarrow} Y$: supposing that Y is MPR of X, this notation indicates the absence of an awaited DATA message generated by X and forwarded by node Y.

For specifying the clauses concerning trust in the protocol, we use the language proposed by [7] which allows to express trust by the fact that if an entity A trusts an entity B in some respect, informally means that A believes that B will behave in a certain way and will perform some action in certain specific circumstances. With this language, the clauses relating to trust in routing operations are expressed with the following notation:

- the expression $A\ trusts_{fw}(Nodes)$ means that A trusts B ($B \in Nodes$) to forward its messages. Otherwise, A not trusting relation is noted by $\neg trusts$,

3 Synthesis of Previous Works

In our previous works [2, 4], we have specified the implicit trust in OLSR, i.e. the trust relationships that should exist between the nodes according to the OLSR protocol. Then we have focused on the detection of attacks on MPR selection, where the attacker abuses the properties of the selection algorithm (HELLO message contents and scheduling) to be selected as MPR. In this section we review the results of these works.

Thus, each node is able to verify the behavior of each neighbor. Such verification can be improved by correlating the information provided by neighbors. First, a node can check the consistency between the HELLO messages of its neighbors:

$$X \overset{HELLO}{\leftarrow} Y, X \overset{HELLO}{\leftarrow} Z,\ Z \in NS_Y, Y \notin NS_Z \Rightarrow X \neg trusts(Y, Z) \tag{1}$$

Second, a node can check the consistency between the HELLO and TC messages of its neighbors:

$$X \overset{HELLO,TC_Y}{\leftarrow} Y, X \overset{HELLO}{\leftarrow} Z, Z \in TC_Y, Y \notin MPRS_Z, \Rightarrow X \neg trusts(Y) \tag{2}$$

Third, a node can check the consistency of MPR selection and routing table of its neighbors and verify that the resulting information they announce is coherent with the locally observed information:

$$NS_A \subseteq NS_B, \exists Z \in TC_A \cap TC_B \Rightarrow X \neg trusts(A, B, Z) \tag{3}$$

It is worth to point out that the mistrust reasonings cannot every time allow the precise identification of the misbehaving node, but allow the detection of inconsistent behavior related to a group of nodes which includes the attacker.

4 Simulation Results

In previous works [2, 4], we have illustrated the effectiveness of trust reasoning for attacks detection using small / trivial attack schemes, where the attacker's position was important to the success of the attack. However, the simulation with large scale ad-hoc networks is necessary to prove the effectiveness and capability of the attack detection, whatever the position of the attacker and the number of nodes in the network. In this section, we discuss the simulation of OLSR with the integration of the previous formulae. We evaluate the effectiveness of our approach under various attack scenarios, and the capacity of mistrust based verifications to identify the attacker nodes.

4.1 Implementation

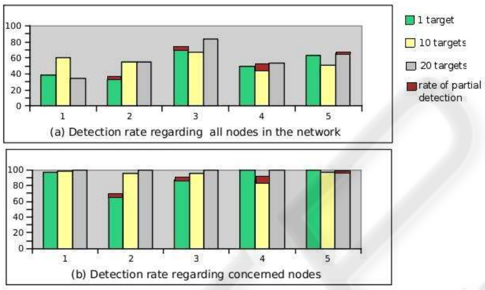
We have used the GlomoSim Simulator and the OLSR patch developed by the Niigata University [1] to simulate the attacks and previous formulae. We have added to this patch a module implementing mistrust rules, and several attack scenarios. In our simulations, ad-hoc networks are composed of different number of nodes (30, 50 and 100 nodes) which are placed randomly. Moreover, the attackers are selected randomly, and each one selects randomly an attack scenario, as well as a set of targets according the selected attack. However, since the ad-hoc networks have to be stabilized to allow the attacker to perform the previous attacks, we have considered that nodes are not mobile. In the following, we discuss only results with 100 nodes using the first attack scenario which takes place according to the following steps:

1. The attacker A identifies target T, its neighbors and 2 hop neighbors.

2. The attacker A detects its common neighbors with the target T, and modifies its HELLO messages to advertise their neighbors as symmetric neighbors as well as an additional fictitious node X.
3. The attacker advertises as its MPR selectors the target's 2 hop neighbors in its TC messages: $\forall Y \in NS_A \cap NS_T, \forall Z \in NS_Y, Z \notin NS_A : TC_A = TC_A \cup Z, X$.

According to OLSR specification, the target has to select the attacker as MPR because it provides reachability for the nodes Z and X, and so the attacker can control some target's flows.

4.2 Results and Discussion



(a) Detection rate regarding all nodes in the network

(b) Detection rate regarding concerned nodes

Fig. 1. Simulation results with 100 nodes.

The diagram (a) in (4.2) presents simulation results of the rate of nodes which are able to detect the attack compared to the total number of nodes in the network. As we can see, the percentage of detection never reaches 100%. However by analyzing the results for each situation, we have deduced that the percentage compared to the total number of nodes in the network was not significant, since the previous formulae do not allow the detection by all the nodes of the network, but only by the *concerned nodes* that are directly or indirectly impacted in the attack scenario (e.g., the target, the nodes used by the attacker to perform the attack and the attacker neighbors).

According to this, in the second step of simulations, we have decided to study the percentage of the concerned nodes that detect the attack. For that, we first have identified the concerned nodes for each attack scenario. Since the concerned nodes depend on each attack scenario, we take the network presented in figure (2) as an example for the first attack. In this scenario, the concerned nodes which have to detect the attack using trust reasoning are :

1. The target node is the first and the most important concerned node, because it is the target of the attack.
2. The target's 2 hop neighbors have to detect the attack because they are advertised as symmetric neighbors and MPR selectors by the attacker when they are not
3. The neighbors of the faulty neighbors advertised by the attacker detect the attack by comparing local information with the attacker TC message.

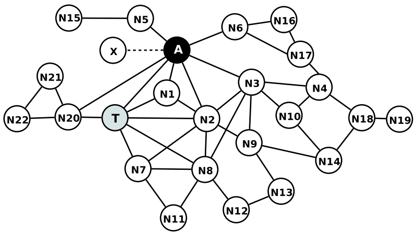


Fig. 2. Network example: A is the attacker, T is the Target.

4. The nodes that can compare the neighborhood of the attacker and the target neighbors (which provide reachability to the 2 hop neighbors of the target advertised by the attacker).

To illustrate the attack detection, we take the example presented in figure 2. The attack takes place according to the following steps :

1. The attacker A identifies target T, its neighbors $\{N1, N2, N7, N8, N20\}$ and 2 hop neighbors $\{N3, N7, N8, N9, N12, N11, N21, N22\}$.
2. The attacker A detects its common neighbors with the target T: $(N1, N2, N20)$, and modifies its HELLO messages to advertise their neighbors ($N7, N8, N9, N21, N22$) as its symmetric neighbors: $NS_A = \{T, N1, N2, N3, N5, N6, N20, \mathbf{N7}, \mathbf{N8}, \mathbf{N9}, \mathbf{N21}, \mathbf{N22}, \mathbf{X}\}$ (X is the additional fictitious node).
3. The attacker advertises as its MPR selectors the target's 2 hop neighbors in its TC messages: $N7, N8, N9, N21, N22 \in TC_A$.

According to OLSR specification, the target T has to select the attacker A as MPR, allowing the attacker to control some target's flows. In this example, the concerned nodes are the target T, the target's 2 hop neighbors $N7, N8, N9, N21, N22$ since they are advertised as symmetric neighbors by the attacker, and the neighbors of these nodes since they are indirectly impacted by the attack (they should be neighbors of the attacker but they are not), and the nodes that are able to correlate the information advertised by the attacker with other information. Using the previous formulae, all the concerned nodes are able to detect the attack:

1. The target node T detect the attack using the following formulae:
   - Formula 1: the node detect inconsistency between HELLO messages of $N7, N8$ and A, where $N7, N8 \in NS_A$ but $A \notin NS_{N7}, A \notin NS_{N8}$.
   - Formula 2: the node detect inconsistency between HELLO messages of $N7, N8$ and TC message of A, where $N7, N8 \in TC_A$ but $A \notin NS_{N7}$ and $A \notin NS_{N8}$.
   - Formula 3: in this example, the node $N9$ will select $N2$ as MPR and nodes $N21, N22$ will select $N20$ as MPR. In the reception of the TC messages of $N2$ and $N20$, the target will detect inconsistency (3), because the neighborhood of $N2$ and $N20$ are included in the attacker neighborhood, and nodes $N2$ and $N20$ should not be selected as MPR.

2. The target's 2 hop neighbors advertised as symmetric neighbors and MPR selectors by the attacker ($N7, N8, N9, N21, N22 \in TC_A$) detect the attack. For instance the node $N7$ detects the attack using the formulae 1 and 2: when it receives the TC message of the attacker, it detects inconsistency because it has not selected attacker as MPR and the attacker is not a symmetric neighbor $A \notin MPRSS_N7$.
3. The neighbors of the faulty neighbors advertised by the attacker detect the attack. For instance the node $N2$ detects inconsistency using the formula 2 in the reception of the TC messages of the attacker because $N7, N8, N9 \in TC_A$ but $A \notin NS_N7, A \notin NS_N8, A \notin NS_N9$.
4. The nodes that can compare the neighborhood of the attacker and the target neighbors (which provide reachability to the 2 hop neighbors of the target advertised by the attacker). For instance when the node $N1$ receives TC messages of the attacker and $N2$, it detects inconsistency using formula 3 because the neighborhood of $N2$ is included in the attacker neighborhood.

The diagrams (a) and (b) in the figure (4.2) allow to compare simulation results regarding total nodes in the network , and concerned nodes by the attack. Simulation results about concerned nodes detecting the attack show that using mistrust reasoning, attacker is detected exactly or partially by all the concerned nodes. Partial detection is the case where a node detects an inconsistency between several nodes, including the attacker, but is unable to determine exactly who is attacking (for example formula 1). Certain nodes can detect the attacker partially and exactly using different formulae (for example formulae 1 and 2). In this case, they can deduce exactly which node is the attacker, and ignore the partial detection. However, partial attack detection provide mistrust information, which can be considered by a node in future cooperation with other nodes to take important decision (for example MPR selection) or correlated with other partial detections in order to deduce exactly the misbehavior node.

The simulation allows us to prove that verifications are a matter of local node behavior producing a global effect on the ad hoc network. Indeed, each node can reason locally on its direct observation to detect inconsistencies without the need for the opinions of other nodes or to cooperate with them. These results reveal the effectiveness of trust-based reasoning for detecting attacks and preventing from the problem of false opinions that can occurs by sharing trust information.

5 Conclusions

Using simulation, we have demonstrated the effectiveness of the verification based on mistrust reasoning in the attack detection. The results allow us to set up verifications that each node can perform to assess the correct behavior of the other nodes and detect attacks against OLSR. It is important to mention that the OLSR protocol (messages) is unchanged, and so our approach still compatible with the bare OLSR.

These results motivate extending the approach for evaluating and distributing the trust information in order to mitigate a partial detection and come to a total detection. Indeed, cooperation between node by sharing trust information could be used to enforce the detection of the misbehavior nodes based on distributed decision. However, it is worth to point out that second-hand information can be subject to false accusations. To mitigate this problem, we plan to set up a mechanism that allows each node to give a proof of its mistrust opinion to participate in the propagation (distribution) of mistrust towards the network. Such a mechanism could be used to enforce a reputation systems by establishing (verifying) trust relationships before cooperating with the other nodes.

References

1. OLSR patch for GlomoSim. http://www.net.ie.niigata-u.ac.jp/mase/olsr/.
2. A. Adnane, R. T. D. Sousa, C. Bidan, and L. Mé. Analysis of the implicit trust within the OLSR protocol. *IFIPTM-2007, Joint iTrust and PST Conferences on Privacy, Trust Management and Security, Moncton, New Brunswick, Canada*.
3. A. Adnane, R. T. D. Sousa, C. Bidan, and L. Mé. Integrating trust reasonings into node behavior in OLSR. *the 3-rd ACM International Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet 2007), Chania, Crete Island, Greece.*, October 2007.
4. A. Adnane, R. T. D. Sousa, C. Bidan, and L. Mé. Autonomic trust reasoning enables misbehavior detection in olsr. *SAC'08 : 23rd Annual ACM Symposium on Applied Computing, Fortaleza, Ceará, Brazil*, March 2008.
5. T. Clausen and P. Jacquet. IETF RFC 3626: Optimized Link State Routing Protocol OLSR. October 2003.
6. M. Wang, L. Lamont, P. Mason, and M. Gorlatova. An effective intrusion detection approach for OLSR MANET protocol. *First Workshop on Secure Network Protocols (NPSec). Boston, Massachusetts, USA*, July 2005.
7. R. Yahalom, B. Klein, and T. Beth. Trust relationships in secure systems – A distributed authentication perspective. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1993.