Modeling and Detecting Intrusions in ad hoc Network Routing Protocols

Mouhannad Alattar, Francoise Sailhan, Julien Bourgeois

Modeling and Detecting Intrusions in ad hoc Network Routing Protocols

Mouhannad ALATTAR
LIFC Laboratory
University of Franche-Comté, France
Email: firstName.lastName@univ-fcomte.fr

Francoise Sailhan
CEDRIC Laboratory
CNAM-Paris, France
Email: firstName.lastName@cnam.fr

Julien Bourgeois
LIFC Laboratory
University of Franche-Comté, France
Email: firstName.lastName@univ-fcomte.fr

***Abstract*—Ad hoc networks mostly operate over open and unprotected environments and are as such vulnerable to a wide range of attacks. Preventive techniques e.g., firewall and encryption, are no longer sufficient and should be coupled with advanced intrusion detection. We propose a distributed intrusion detection system that analyses activity logs so as to generate the rules which are used to detect intrusion. In order to deal with the distributed nature of an *ad hoc* network, the proposed system correlates information found in the multiple traces provided by surrounding devices. Performance is further evaluated, in terms of e.g., intruder detection rate and false positive.**

I. Introduction

Securing *ad hoc* networks is not a trivial task because these networks rely on open radio-based medium of communication. In addition, the lack of centralized points e.g., switches and routers, complicates the deployment of preventive strategies. Thus, traditional ways of securing networks e.g., firewall, should be enriched with reactive mechanisms including intrusion detection system. As a first step upon detecting intrusions, we first categorize the attacks threatening routing protocols; their central role, i.e., determining multi-hops paths among the devices, designates these latter as the favourite target of attackers. Then, we propose a Distributed Intrusion Detection System (DIDS), which analyses the activity logs so as to discover a sequence of events that characterizes an intrusion attempt. Then, logs are correlated so that the DIDS correctly identifies more intrusions and reduces the number of false positive. We further examplify and experimentally evaluate the performance of our DIDS focusing on a specific attacks, the so-called spoofing link attack, which aims at undermining the Optimized Link State Routing (OLSR) protocol [6].

The remainder of this paper is organized as follows. We first present the attacks on *ad hoc* routing protocols (§II). Then, we envisage the development of the proposed DIDS (§III) and its evaluation (§IV). Finally, we conclude this article presenting future research directions (§V).

II. Attacks on OLSR

In *ad hoc* network, routing protocols constitute one of the favourite target of intruders; the reason is threefold. First, no security countermeasure is specified as a part of the RFCs proposed by IETF working group[1]. Second, the absence of the centralized infrastructure complicates the deployment of preventive measures e.g., firewalls. Third, any device may operate as router, which facilitates the manipulation of multi-hops messages as well as the compromising of the routing functionality. Attacks targeting *ad hoc* routing protocol fall into two main categories, passive (i.e., observing the traffic) *versus* active (i.e., an authorized change is attempted). Active attacks are further sub-classified according to the undertaken action on the routing messages [3]:

- *Drop attacks* consist in dropping one (or further) routing message(s).
- *Modify and forward attacks* modify received routing message(s) before forwarding it.
- *Forge reply attacks* aim at sending false response(s) to routing message(s).
- *Active forge attacks* proactively generate novel routing message(s).

In this paper, we focus on a particular active forge attack corresponding to a spoofing link attack. We exemplify this attack using the OLSR protocol [6]. In a nutshell, OLSR aims at maintaining a constantly updated view of the network topology on each device. One fundamental of OLSR is the notion of multipoint relay (MPR): a subset of 1-hop neighbors that covers all the 2-hops neighboors and forwards the control traffic in the entire network. In practice, a node recognizes the 1-hop neighboors through periodic heartbeat messages, termed Hello messages. A MPR declares the nodes that selected itself to act as MPR via *Topology Control* (TC) message, which is intended to be diffused in the entire network. Thanks to TC messages, any device computes the shortest path (in term of number of hops) to any destination, such path being represented as a sequence of MPRs. Overall, OLSR is subject to a variety of attacks, including spoofing link attack, that majoritly target MPRs ; these latter constituing an attractive post for launching further attacks (e.g., deleting messages). In practice, one possible strategy consists in corrupting the MPR selection.

A. Spoofing link attack

An attacker aspiring being selected as an MPR may corrupt the MPR selection (equation 1). Towards this goal, the intruder advertises a falsified local topology: I sends a Hello message ($V \xleftarrow{Hello(NS'_I)} I$) to a victim V so as to declare a neighboor set NS'_I differing from the real set NS_I. The difference results from inserting[2] (i) a non-existing node N ($N \notin \mathcal{N}$ with $\mathcal{N}$ defining the set of nodes composing the

[1] http://www.ietf.org/dyn/wg/charter/manet-charter.html

[2] A falsification of the local topology may also consists in suppressing existing neighboor(s) in the advertised neighbooring set. Nevertheless, rather than facilitating the selection of the intruder as MPR, this alteration reduces the connectivity of the intruder perceived by other and hence mortgages the probability of being selected as MPR.

network) and/or (ii) an existing, but non-neighbooring, node E ($E \in NS'_I \cap \mathcal{N} \ni: E \notin NS_I$). Thus, the connectivity of I is increased $Card(NS'_I \backslash NS'_I \cap NS_I) > 0$. Recall that the set of MPRs is selected so that all the 2-hop neighboors are covered, I is hence selected as a MPRs of V. This affirmation is verified as long as no other intruder I' (respectively legitimate node M) advertises the same neighboor N (respectively E) .

$$
\begin{array}{c}
I \xleftarrow{Hello(NS_V)} V, V \xleftarrow{Hello(NS'_I)} I, \\
(\exists N \in NS'_I \ni: N \notin \mathcal{N} \cap NS_I) \vee (\exists E \in NS'_I \cap \mathcal{N} \ni: E \notin NS_I) \\
\Downarrow \qquad\qquad\qquad\qquad \Downarrow \\
Card(NS'_I \backslash NS'_I \cap NS_I) > 0 \qquad Card((NS'_I \backslash [NS'_I \cap NS_I]) \cap \mathcal{N}) > 0 \\
\exists M \in \mathcal{N} \backslash \mathcal{I} \ni: M \in NS_V \wedge E \in NS_M \qquad \exists I' \in \mathcal{I} \ni: I' \in NS_V \wedge N \in NS_{I'} \\
\Downarrow \qquad\qquad\qquad\qquad \Downarrow \\
I \vee M \in MPR_V, \qquad\qquad I \vee I' \in MPR_V, \\
\Downarrow \\
I \in \mathcal{I}.
\end{array}
\tag{1}
$$

Such an attack can be detected relying on a intrusion detection system.

III. Intrusion Detection

We proposed a DIDS that traces and detects the source of a network-based intrusion. This DIDS includes a host-based tracing mechanism that keeps track of the OLSR activities (routing logs) and analyses these latter to detect evidences. Each device further cooperates with one another so as to correlate evidences and match it against predefined intrusion signatures. In spoofing link attack, recall that the intruder increases artificially its connectivity so as to be chosen as MPR, we define three evidences that render a MPR suspect:

- a MPR relays packets in an abnormal way e.g., packets are dropped and do not reach their destination,
- a MPR is replaced suspiciously by a new MPR,
- a MPR is the only one covering one node.

If one of the above evidence is discovered, advanced correlation and investigation is performed: messages are exchanged with the 2-hop neighboors that are covered by the suspicious MPR.

IV. Performance evaluation

In order to evaluate our DIDS, we couple a network simulator (NS3)[3] [5] with LinuX Containers virtual machines (LXC)[4] [4]. NS3 simulates a MANET (and hence owns an implementation of the OLSR protocol) while each simulated devices owns a LXC container embeding a DIDSs. While offering the capability of monitoring the memory consumption of each node in the virtual machine, this platform permits to easily experiment a MANET (herein 25 mobile nodes including 5 intruders are simulated). Performance is further evaluated in terms of intruder detection rate, resource consumption and false positive (Figure **??**). The detection rate rises up to 95% with a node's speed equal to 10 m/s. Increasing the speed leads to a decrease of the detection rate that reaches 55% with a node speed of 20 m/s (i.e., 180 km/h). This decrease results from the difficulty to obtain investigation correspondences when the devices mobility rises. In counterpart, the number of false positive observed is limited (ranging between 0 up to 3) and has almost no relation with the mobility. The network overhead resulting from DIDS detection messages is 10 times smaller than the overhead attributed to the OLSR protocol. Furthermore, our system causes an increment in the container used memory ranging from 16 to 52 MB, meaning that such a DIDS can be deployed on resource-constrained devices.

Overall, our system has an accepted (respectively high) intruder detection rate in high (respectively moderate) mobility network.

[3]http://www.nsnam.org

[4]http://lxc.sourceforge.net

V. Conclusion and Future work

In this paper, we present a distributed intrusion detection system that is focusing on the network intrusions targeting the routing protocol in MANET. The proposed DIDS reffers to a host-based detection system that matches logs against intrusion signatures. It requires no prior knowledge about the *ad hoc* network. Therefore, our system has an advantage over anomaly-based IDSs, which search for deviations from a normal expected behavior (and hence is generated from training data) so as to detect the intrusions. In counterpart, our system cannot detect a not-defined intrusion. DIDS performance is investigated against spoofing link attack. The detection rate reaches up to (95%) with a mobility speed more than the running average speed of a human being. The number of false positive is limited. Even though the amount of logs to be analyzed increases exponentially with network size, resource consumption is still suitable for the resource-constrained devices.

References

[1] F. Sailhan and J. Bourgeois, *Log-based dis-tributed intrusion detection for hybrid net-works*, CSIIRW '08. New York, NY, USA, 2008.

[2] F. Sailhan, J. Bourgeois and V. Issarny, *Security supervision system for hybrid networks*. Springer, first edition, 2008.

[3] N. Peng and S. Kun, *How to misuse aodv:a case study of insider attacks against mobile ad-hoc routing protocols*. Elsevier Science Publishers B. V., 2005.

[4] Sukadev Bhattiprolu and al., *Virtual servers and checkpoint/restart in mainstream linux*. SIGOPS Oper. Syst. Rev., 2008.

[5] George F. Riley and Thomas R. Henderson, *The ns-3 network simulator*, In Modeling and Tools for Network Simulation. Springer Berlin Heidelberg, 2010.

[6] T. Clausen and P. Jacqueti, *Optimized link state routing protocol (olsr)*. IETF experimental RFC 3626, october 2003.