Combining Passive Autoconfiguration and Anomaly-based Intrusion Detection in Ad-hoc Networks

Stephan Schuhmann
Institut für Parallele und Verteilte Systeme (IPVS)
Universität Stuttgart
Stuttgart, Germany

Lars Völker
Institut für Telematik
Universität Karlsruhe (TH)
Karlsruhe, Germany

Abstract

Two essential services of Ad-hoc networks are IP address autoconfiguration and intrusion detection systems. Since both autoconfiguration and intrusion detection may base their decisions on routing protocol anomalies, their interdependencies can cause problems. In this paper, we present our approach to efficiently combine autoconfiguration and intrusion detection, and present our enhancements in attack detection for an autoconfiguration system. We have identified anomalies indicating specific attacks, implemented suitable anomaly detectors, and evaluated our system. The results show that it is possible to detect both the attacks and IP address conflicts in an efficient way.

1. Introduction

Ad-hoc networks can be used to easily deploy wireless networks when an infrastructure is missing. An important aspect of these networks is the automatic configuration of IP addresses. Passive autoconfiguration systems like PACMAN [13] observe the behavior of the ad-hoc routing protocols in order to detect anomalies caused by misconfigured IP-Addresses and reconfigure the network, accordingly. A state of the art intrusion detection systems (IDS) also looks for anomalies in the routing protocols but aims to identify attacks. Unfortunately, the approaches taken by both systems are very similar and in conflict with each other. Packets might be classified by the IDS as an attack, while the autoconfiguration system only detects duplicated addresses. Furthermore, both systems would do similar classification work on the routing protocol messages and therefore be inefficient.

In this paper, we examine how both approaches can be combined in an efficient way. We chose PACMAN as an autoconfiguration system and extended it to also function as an intrusion detection system. Using Optimized Link State Routing (OLSR, [4]) as an example, we examined attacks against the routing protocol and identified the anomalies such attacks cause. We extended PACMAN with a common detection function which detects anomalies of the routing protocols and allows us to interpret anomalies for autoconfiguration and intrusion detection.

The paper is structured as follows: After this introduction, we present PACMAN, the OLSR routing protocol, and related work. Subsequently, attacks against OLSR and occurring anomalies are presented and classified in Section 3. In order to detect these anomalies, several algorithms have been developed and are presented in Section 4, while Section 5 shows evaluation results. The paper closes with Section 6 that summarizes the entire work and gives an outlook on possible future work.

2. Fundamentals

PACMAN consists of routing protocol specific functionality like parsers and global functions like “Address Assignment”, “Address En-/Decoding”, “Address Change Management”, “Conflict Resolution”, and “Passive Address Resolution”. In addition, the “Passive Duplicate Address Detection (PDAD)” exists—the part of PACMAN, which identifies the anomalies needed to detect the address changes. This functionality is very similar to an anomaly-based intrusion detection system; hence, we introduce many changes and new functionality here.

PACMAN supports several MANET routing protocols such as the widely-used ad-hoc on Demand Distance Vector Routing (AODV, [10]) and Optimized Link State Routing (OLSR). Our work is based on OLSR but our approach can work with other routing protocols (e.g., AODV) also. One only needs to identify the routing protocol’s anomalies of possible attacks.

In the paper at hand, we focus on the HELLO and Topology Control messages of OLSR. Other messages, Host Network Address and Multiple Interface Declaration, also al-

Published in Proceedings of the 8th IEEE International Workshop on Applications and Services in Wireless Networks (ASWN '08), pp. 87-95. Kassel, Germany, October 2008.

http://dx.doi.org/10.1109/ASWN.2008.14

low for a few attacks, but are commonly not considered as relevant.

2.1. Related Work

Adjih et al. [2] describe several known attacks against OLSR. These attacks are the point of investigation in this paper. But while this paper tries to secure OLSR by using an IDS, Adjih et al. developed a distributed key management. Many other authors [6], [7], [11] also propose cryptographic methods such as digital signatures or hash-chains to secure MANET routing protocols. Secure OLSR [8] is an exemplary protocol that is based on such methods, therefore being a fundamentally different and complementary approach to ours. While securing an Ad-hoc routing protocol using cryptography alone can be very successful, these solutions often introduce major disadvantages: cryptography is computationally expensive, especially on small mobile devices, and the secured protocols are not that "Ad-hoc" anymore, since trust relationships have to be built.

Orset and Cavalli [9] use a combination of deontic and temporal logic to specify the correct behavior of an OLSR node and describe security policies to prevent attacks on the OLSR protocol, leading to another way to detect routing protocol attacks.

Securing MANETs by means of an intrusion detection system is presented in works from Cabrera et al. [3], Dhillon et al. [5], and Wang et al. [12]. These works have developed, implemented, and successfully evaluated intrusion detection systems, respectively.

While previous work only focused on security, our IDS is a security extension for the PACMAN architecture. Thus, it combines passive autoconfiguration with anomaly-based intrusion detection, providing an easy passive and secure way to configure Ad-hoc networks from the start.

3. OLSR Protocol Anomalies

This section presents the OLSR Protocol Anomalies we identified for known attacks. Following [2], these attacks can be divided into two categories: *incorrect generation* and *incorrect forwarding* of OLSR messages. Common attacks include Replay Attacks, Topology Manipulation Attacks, Blackhole Attacks, Wormhole Attacks, MPR Flooding Attacks, and Address Spoofing—most of which are detected by PACMAN as address conflicts. In order to detect these attacks, one has to identify the anomalies caused by each attack.

The most important anomalies include the following:

Sequence Number Anomalies (SNA) indicate that a message contained a sequence number that was not expected. The following cases can be detected by nodes when receiving at least two packets of the same sender address:

- The sequence number is incremented too fast.
- The sequence number is decremented.
- Two messages of the same sender with different content share the same sequence number.
- A node receives a message with its own source address but a sequence number that is too high.

These anomalies are commonly seen in case of address spoofing and are also used in PACMAN to detect address conflicts. Furthermore, replay attacks and sequence number attacks can be detected using this anomaly.

Table 1. Survey of routing protocol anomalies

Anomaly	Full Name	Description
SNA	Sequence number anomaly	Reception of message with unexpected sequence number
INC_NEI	Inconsistent Neighbor Sets	Reception of message with inconsistently proclaimed neighbor set
MPR_VIO	MPR Principle Violation	Incorrectly forwarded message by Non-MPR
H_SIP	HELLO Self IP	HELLO message with own IP address as source received
T_SIP	TC Self IP	TC message with own IP address as source received
INC_TC	Incorrect own TC links	TC message with incorrectly proclaimed link to oneself received
NON_TC_L	Nonexistent TC links	TC message with incorrectly proclaimed link to other node received
MIS_FW_TC	Missing forwarded TC message	MPR neighbor did not forward TC message
WILL_VIO	Willingness violation	Forwarded TC message though Willingness = 0 proclaimed

Inconsistent Neighbor Sets (INC_NEI) occur when the proclaimed neighborhood fails sanity checks. This can happen when the attacker spoofs addresses, replays routing protocol messages, or tries to manipulate the topology.

MPR Principle Violations (MPR_VIO) are detected when nodes forward messages without being a MPR. Usually the attacker tries to manipulate the topology. Also in rare cases address spoofing can take place.

HELLO Self IP (H_SIP) denotes that a HELLO message with the own address as sender was received. This is a clear violation since HELLO messages must not be forwarded. This anomaly occurs when the attacker incorrectly forwards messages or spoofs the address. PACMAN uses a similar anomaly detector (Self-Address).

Incorrect Own TC Links (INC_TC) appear when a Topology Control message was received but claimed an incorrect link to the receiving node. This happens when the attacker tries to manipulate the topology.

Nonexistent TC Links (NON_TC_L) indicate again an incorrect Link in a Topology Control message. However, this time the incorrect link is not to the receiver but to a different node.

Missing Forwarded TC Messages (MIS_FW_TC) show that the MPR did not correctly forward a Topology Control message.

Willingness Violations (WILL_VIO) are detected when another node forwards Topology Control messages but proclaims no willingness to do so.

Table 1 summarizes these anomalies. In Section 5.2.2, we show which anomalies were detected by our implementation for specific attacks.

4. Implementation in PACMAN

This section deals with the technical realization of the IDS and gives some implementation details.

In order to detect the anomalies explained in Section 3, nine anomaly detection algorithms have been designed and implemented into PACMAN's PDAD module. Thus, PACMAN simply uses the detection algorithms as additional module besides its own modules that aim to detect duplicate addresses in a passive way. Each module is able to detect a specific type of anomaly. As OLSR implementation we used olsrd-0.4.10 from `olsr.org` [1].

For every incoming OLSR HELLO or TC routing message, a node checks the detection algorithms within PACMAN's PDAD module at first to verify if this message causes a protocol anomaly to occur. Afterwards, the node possibly processes the message, dependent on its MPR status. If an algorithm detects an anomaly, it increases a global anomaly counter. This leads to special combinations of anomalies which arise under particular attacks, as shown later in this paper.

Some of the algorithms require comparison of protocol fields with those of previously sent or received messages, for example to detect oscillating link sets in link spoofing attacks. Therefore, additional space is needed to allocate tables, which maintain this history. The overhead per node for storing a table that contains the last 256 incoming and outgoing messages, respectively, is about five kilobytes, as one single entry takes 10 bytes in average for storing the relevant protocol fields like addresses, hop counters, or included link sets. This overhead does not have great impact regarding memory constraints even on small devices like cellular phones.

Listing 1 presents the "Nonexistent TC Links" PDAD module in C-like pseudo code as an example to clarify how our system works. This anomaly occurs in link spoofing attacks at the victim of the attack. In this module, the anomaly detection algorithm of a victim Y of the attack that receives this message compares all announced link sets proclaimed in an incoming TC message with its own links. If the message includes Y's primary inferface's address, this node checks if the originator of this message is included in its link set. As links are assumed to be symmetric by OLSR, the inclusion of this link is a clear protocol anomaly and can only happen when the link is initially established.

```
if (msg->type == ”TC_MESSAGE”) {
 for (int i = 0; i < msg->link_set.getSize(); i++) {
  if (msg->link_set[i].getAddress() == own_main_addr) {
   int conflict = 1;
   for (int j = 0; j < own_link_set.getSize(); j++) {
    if (own_link_set[j].getAddress() == msg->orig_addr)
     conflict = 0;
      }
   }
    if (conflict == 1) {
      return 1;
    }
 }
return 0;
```

Listing 1. PDAD module for "Nonexistent-TC-Links"

In case of a detected anomaly, the module returns 1, otherwise, it returns 0. Every time the module returned 1 our system will increment the NON_TC counter by one.

The IDS simply consists of additional PDAD modules and a decision logic. This logic determines, based on thresholds for the anomalies (see Section 5.1.3), if an attack is currently taking place. It is easily possible to deactivate several of PACMANs or the IDS basic modules, or even the whole IDS. Hence, our solution is flexible and enables complete separation of PACMAN's and the IDS' functionality, but also the cooperation of the two systems or even parts of them.

The IDS has been implemented as a prototype as well as in the GloMoSim [14] simulator to prove scalability of our approach. The code overhead of our IDS is about 42.4 kilobytes, which represents about 39 % of the original PACMAN code. We consider this overhead as acceptable.

5. Simulation and evaluation results

The implemented IDS has been tested with several attack scenarios where different numbers of nodes were chosen to be malicious. Section 5.1 illustrates the simulation results of larger-scale networks, while Section 5.2 presents the results of the prototype implementation.

5.1. Simulation results

Numerous iterations of our simulation have been run in GloMoSim, using the simulation parameters described in the appendix. The number of direct neighbors of a node varied between seven and 20 nodes, dependent on the location of the node within the network. Sending periods were the same as in the prototype runs, leading to 300 generated HELLO messages and 120 generated TC messages per node.

Regarding simulation runs of the attacks presented in Section 3, the simulation confirmed the results of our real-world evaluation, i.e. the same anomalies that are listed in Table 3 arose. Due to the limited space, we now focus on two specific, exemplary attack scenarios: a *combined IP spoofing/sequence number attack* as an example for incorrect generation of messages, and a *blackhole attack* as an example for incorrect forwarding of control messages. Furthermore, we set thresholds for the anomalies to filter out false positives, as described in Section 5.1.3. The denoted numbers of observed anomalies represent the sums of received original control messages and forwarded messages; thus, probably including duplicates.

5.1.1. Example 1: Combined IP Spoofing/Sequence Number Attack

We simulated two different attacks, dependent on the number of hops between the attacker and the victim of the attack.

In the first simulation run, a single attacker (node X) spoofed the IP address of a direct neighbor, node Y. Thus, X generated HELLO and IP messages which contained the IP address of Y as source address, but at the same time, these messages contained an increased sequence number so that other nodes regard Y's messages as aged due to their lower sequence number. However, X proclaimed its own neighborhood within the HELLO and TC messages. Figure 1 shows the position of X and Y within the grid. In these figures, the edges of the grid represent the node positions.

During simulation time, Y received 32617 TC messages and 3347 HELLO messages. Y detected following numbers of anomalies:

- 811 **SNA** anomalies due to received messages with its own IP address as source address, but too high sequence numbers. This represents about 2.26 % of all received messages.
- 1994 **INC_TC** anomalies due to reception of TC Messages with incorrectly proclaimed links *to oneself* (Y), i.e. 6.11 % of all received TC messages.
- 5851 **INC_NEI** anomalies due to reception of TC Messages with incorrectly proclaimed links *to other nodes*, i.e. 17.94 % of all received TC messages.
- 409 **MPR_VIO** anomalies due to reception of messages forwarded by a non-MPR node, i.e. 1.25 % of all received TC messages.
- 283 **H_SIP** anomalies due to reception of own HELLO messages, i.e. messages with the own IP address proclaimed as source address. This amount represents 8.46 % of all received HELLO messages.
- 8137 **T_SIP** anomalies due to reception of own TC messages, i.e. 24.95 % of all received TC messages.

When using the thresholds as presented later in Table 2, the system correctly detects an ongoing IP spoofing attack at node Y—the victim of this attack.

The remaining anomalies denoted in Table 3 were detected less than 20 times, respectively. They represented false positives at the beginning of the observation time due to engaging effects during the routing protocol's startup phase.

Besides the aforementioned anomalies detected by Y, there arose two types of anomalies which could also be detected by other nodes than Y:

- Sequence number anomalies (**SNA**) arose at some nodes around the victim Y. The distribution of those anomalies is shown in Figures 1 (2-dimensional[1]) and 2 (3-dimensional). Regarding the 2D plot, one can see that Y has selected three MPRs: M_1, M_2, and M_3 which is also an MPR of X. Thus, M_3 does not forward Y's messages as M_3 regards them outdated because it also receives messages from X. So, M_1 and M_2 are the only nodes that forward Y's messages, while all of the other nodes regard them as outdated

[1] The circles represent the nodes' transmission ranges.

and do not forward them. Thus, the nodes that detect the **SNA** anomalies are all the 1 hop neighbors of Y and those 2 hop neighbors of Y, which are reached via M_1 or M_2, as it is shown in Figure 1.

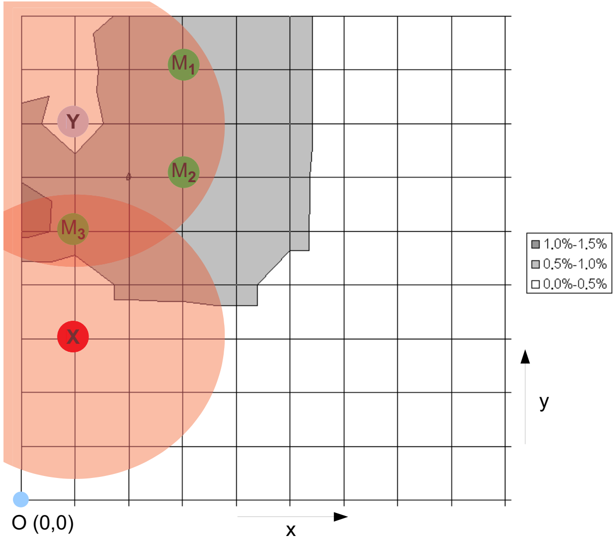


Figure 1. Distribution of SNA anomalies in Example 1, relative to totally received messages per node (2D plot).

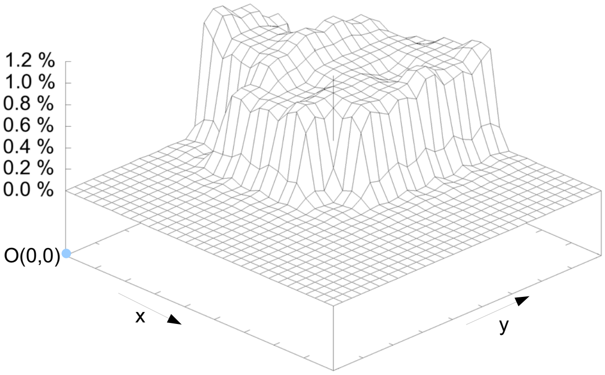


Figure 2. Distribution of SNA anomalies in Example 1, relative to totally received messages per node (3D plot)

- Inconsistent neighbor set (**INC_NEI**) anomalies occured at every single node. These anomalies arose because of different neighbor sets that were proclaimed by X and Y, which carried the same origin address. The amount of anomalies depends on the position of the nodes within the grid and, thus, the total number of received messages, i.e., nodes with a central position in the grid received more messages and, hence, also more anomalies. Figure 3 shows the **INC_NEI** anomaly distribution.

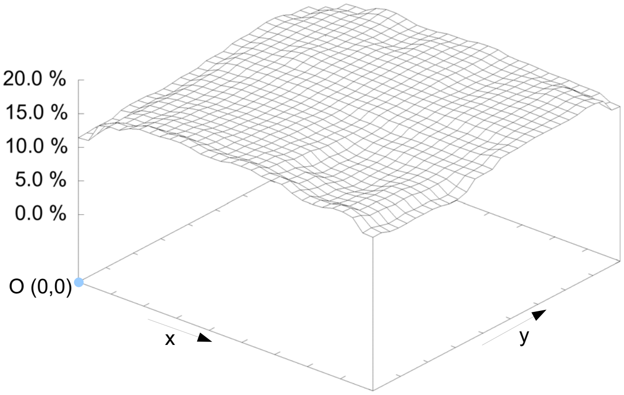


Figure 3. Distribution of INC_NEI anomalies in Example 1, relative to totally received messages per node

5.1.2. Example 2: Blackhole Attack

The second exemplary attack is the blackhole attack with multiple attackers. As mentioned before, the blackhole attack is a simple passive attack. The attacking nodes are elected as multipoint relay by other nodes and should therefore forward TC messages; however, the attacking nodes do not forward TC messages. In order to do so, the attackers at first proclaim a high willingness to forward other nodes' messages within their HELLO packets. This leads to the election of the attackers as multipoint relays. As multipoint relays, they do not forward messages and avoid the distribution of other nodes' neighborhood information within the whole network.

We decided to choose multiple attackers to show that our system also works for more than one attacker. In our particular example, three nodes called X_1, X_2, and X_3 represent attackers that act in the previously described way. The scenario and the arising anomalies are shown in Figure 4 in a 2D plot, and also in Figure 5 in a 3D plot. The figures show that this attack caused **MIS_FW_TC** anomalies at all of the attackers' MPR selectors, as these nodes do not receive forwarded TC messages of the respective nodes and, thus, detect the protocol-inconform behavior of the attackers. The figures also clarify that the relative number of anomalies increases if a node is MPR selector of more than one attacker. The proportion of detected anomalies in this blackhole attack is significantly higher than in the combined attack presented in Example 1 since the direct victims of this attack (the MPR selectors of the attackers) are 1-hop-neighbors of the attackers and detect every omitted forwarding of their

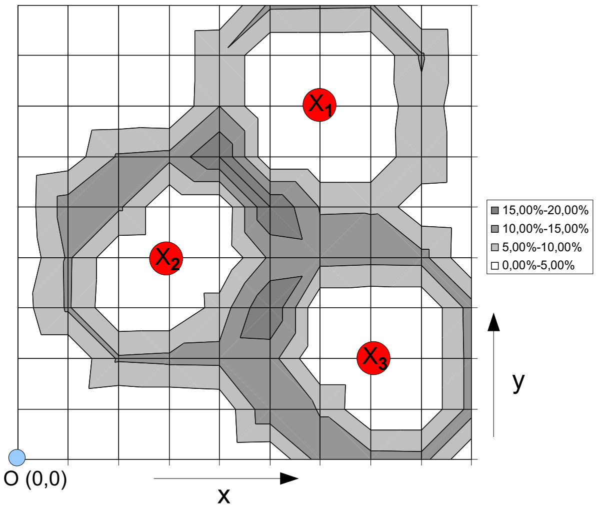


Figure 4. Distribution of MIS_FW_TC anomalies in Example 2, relative to totally received messages per node (2D plot)

own TC messages since they do not receive forwarded messages of X_1, X_2, or X_3.

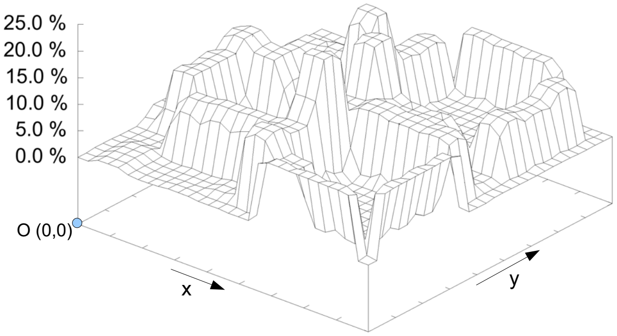


Figure 5. Distribution of MIS_FW_TC anomalies in Example 2, relative to totally received messages per node (3D plot)

5.1.3. Threshold determination for the arisen anomalies

After we simulated all of the attacks described in Section 3, we tried to obtain thresholds for the arisen anomalies in terms of the relative amount of anomaly-afflicted received messages compared to the total number of received messages. The thresholds for the distinct anomaly detection algorithms are used to filter out the false positives. At the same time, we avoided to choose thresholds too high, keeping the IDS sensitive enough.

Table 2. Defined thresholds

Anomaly	Minimum #True Positives	Maximum #False Positives	Threshold
MIS_FW_TC	5.43 %	0.39 %	2.91%
INC_NEI_SET	11.42 %	0.89 %	6.15%
MPR_VIO	0.25 %	0.02 %	0.14%
SNA	0.69 %	0.00 %	0.35%
H_SIP	15.56 %	0.00 %	7.78%
T_SIP	23.33 %	0.00 %	11.67%
INC_TC	9.56 %	0.00 %	4.78%
NON_TC_L	9.26 %	0.02 %	4.64%
WILL_VIO	26.67 %	0.00 %	13.33%

Table 2 displays the thresholds we determined for the various anomalies. We determined the thresholds as arithmetic means of the minimal proportion of anomaly-afflicted received messages at a specific node and the maximum proportion of false positives at a node that is not affected by an attack. Values above a certain threshold denote that the system assumes that an attack is taking place, and values below are filtered out as false positives.

5.2. Prototype results

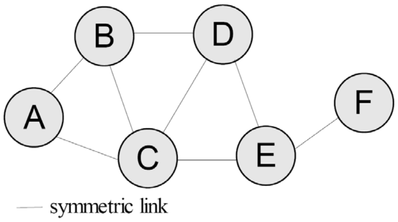
The implemented IDS prototype has been evaluated on six x86-PCs. The attacks presented in Section 3 have been investigated on different topologies. In order to determine false positives and arising anomalies during the phase when the routing protocol was started on all nodes, scenarios in absence of attackers have been analyzed. Unless otherwise noted, all of the described scenarios in this paper have been evaluated over a timespan of 30 minutes with the following sending periods for every node:

- HELLO period: 2 seconds
- TC period: 5 seconds

Thus, every node has sent 900 HELLO and 360 TC messages within the evaluation period.

5.2.1. Attacker-free scenarios

To determine the amount of false positives (detected anomalies although no attack takes place), scenarios with properly acting nodes have been conducted. The corresponding topology is shown in Figure 6. Using this topology, we evaluated the number of false positive in two ways that solely differed in the observation time. The first scenario had a regular runtime of 30 minutes, while for the



Node	IP Address
A	192.168.0.1
B	192.168.0.2
C	192.168.0.3
D	192.168.0.4
E	192.168.0.5
F	192.168.0.6

Figure 6. Scenario for determination of false positives

second scenario, this time was limited to one minute to find out how many of the anomalies from the first scenario arose during the topology finding phase of the network, e.g., anomalies that arise because of empty routing tables.

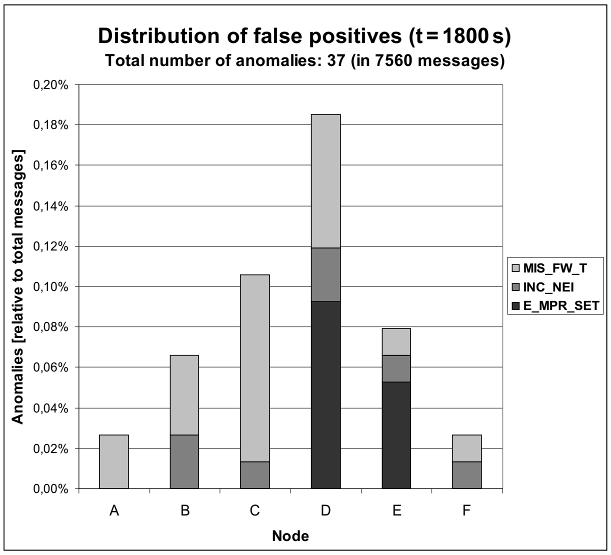


Figure 7. False positives at scenario with runtime t_1 = 1800 s

Figure 7 shows the numbers of anomalies (relative to the total message amount) that have been detected by the particular nodes. For clarity reasons, anomalies that did not appear are left out in this figure. Regarding the first scenario, a total number of

$$6 \cdot (900 + 360) = 7560 \text{ messages}$$

have been produced and sent by the six network nodes. Thus, only a maximum of $37/7560 \approx 0.5$ %[2] of these messages become false positives, and $32/37 \approx 86.5$ % of these anomalies arose during the first 60 seconds of observation, as Figure 8 shows. This shows clearly that especially at long-term observations with considerably less than 1 % of the sent messages led to anomalies, our IDS performs very well concerning false positives. The number of anomalies observed during the startup phase can be reduced if a node starts the anomaly-detection algorithms not until its tables for incoming and outgoing messages are filled up to a specific amount.

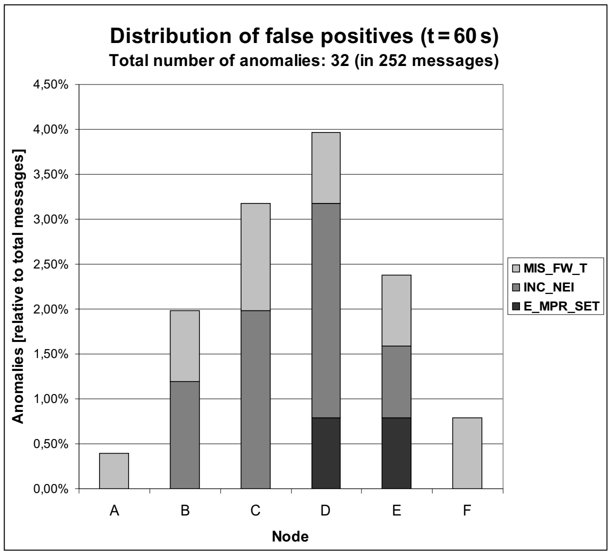


Figure 8. False positives at scenario with runtime t_2 = 60 s

It has to be mentioned that a properly acting node which accidentally configures an address already in use also leads to false positives as our IDS detects a duplicated address. However, as PACMAN detects and mitigates this address conflict, the anomalies will only be observed for a very short time and result in an irrelevant number of false positives.

5.2.2. Attacker scenarios

Different attacks have been implemented and performed to detect which anomalies occur at specific nodes.

IP spoofing attacks have been performed on nodes with different hop distances. In all of these scenarios, sequence number (**`SNA`**), MPR principle violation (**`MPR_VIO`**) and incorrect neighbor set (**`INC_NEI`**) anomalies occured at every network node because of distinct neighborhoods of the attacker and the victim of the attack. Furthermore, the victims of the attack detected TC messages which were proposed to be of themselves (**`T_SIP`**), but carried nonexistent links within. The victims also detected messages with their own address as source address which is not possible without

[2] The actual value should be lower since some messages triggered more than one anomaly.

Table 3. Results of attacker scenarios

Attack		**Anomalies, occuring at...**		
		...all nodes	...all direct neighbors of attacker	...victim of attack
IP spoofing	1-hop-spoofing	SNA, INC_NEI, MPR_VIO	(none)	H_SIP, T_SIP, INC_TC
IP spoofing	≥ 2-hop-spoofing	SNA, INC_NEI, MPR_VIO	(none)	INC_NEI, NON_TC_L
Link spoofing (proclaimed nonexistent links)		INC_NEI	(none)	NON_TC_L
Sequence number attack		SNA	(none)	(none)
Blackhole attack		(none)	MIS_FW_TC	(none)
Replay attack		SNA, INC_NEI	(none)	INC_TC
Wormhole attack (collaborating attackers)		(none)	(none)	H_SIP
MPR flooding	Proclaimed willingness of zero	(none)	WILL_VIO, MPR_VIO	(none)
MPR flooding	Proclaimed empty MPR Set	(none)	E_MPR_SET, MPR_VIO	(none)

malicious nodes since the source address is changed hopwise, as well as HELLO messages with their own address as source address (**H_SIP**) which is impossible since HELLO messages are never supposed to be forwarded.

The **declaration of nonexistent symmetric links** in TC messages caused inconsistent link sets for all nodes (**INC_NEI**) since protocol-conform link partners do not proclaim these links. Omission of existent links in TC messages became a problem for our IDS since it did not detect any anomalies. The difficulty here is to distinguish between omission of existent links and links that became asymmetric because of different transmission ranges since they lead to the same impact for OLSR, which relies on symmetric links: it makes those links unusable for routing.

Sequence number attacks led to diverse anomalies in the proclaimed sequence numbers of attacker and victim of the attack (**SNA**) such as non-increasing or duplicated sequence numbers, dependent on the proclaimed sequence number. All of the cases described in Section 3 have been tested and caused at least one detection algorithm to indicate anomalies.

In case of a **blackhole attack**, the neighbors of the attacker did not receive any forwarded messages from this node (**MIS_FW_TC**). Since the attacker proclaimed a high willingness in order to be elected as MPR and to make this attack effective, this denial of message forwarding was a clear protocol anomaly.

Replay attacks caused anomalies in sequence numbers (**SNA**) and proclaimed neighbor sets (**INC_NEI**) since the attacker forwarded aged TC messages, as well as inconsistent TC messages (**INC_TC**).

If a single attacker performed a **wormhole attack**, the nodes amongst which a wormhole was created detect anomalies since they receive their own HELLO messages (**H_SIP**). This would not be possible if every node acts protocol-conform and hence does not forward any HELLO messages. Furthermore, the nodes which are neighbored to an attacker and the corresponding victim of the attack receive the same HELLO message twice (**MUL_H**). This is a second anomaly.

If multiple attackers create a wormhole by tunneling HELLO messages from distinct nodes, only victims detect the above mentioned anomaly of receiving their own HELLO messages (**H_SIP**).

In our test scenarios, **MPR flooding attacks** could be detected by the neighbors of the attacker.

As a comparison, it can be said that with the exception of omitted links in a link spoofing attack, no false negatives have appeared. The omission of links only influences the links to the attacker and does not impact correctly behaving nodes significantly. All of the other attack variants caused different combinations of anomalies.

6. Conclusion

We have presented a passive, anomaly-based way to detect intruders in Ad-hoc networks. In combination with PACMAN, the architecture in which our IDS is embedded, an easy method to autoconfigure MANETs in the presence of malicious nodes was developed. Different scenarios have been tested in a prototype implementation and in GloMoSim simulations, showing very good performance in the detection of anomalies arising at particular attacks, as in the scenario we analyzed for this paper the number of false positives was considerably lower than the number of true positives and, thus, the definition of thresholds for the various anomalies was possible. The thresholds are used to determine if a detected attack currently takes place.

Nevertheless, an open issue is the detection of anomalies in dynamic scenarios with mobile nodes since we focused on static scenarios. Some preliminary simulations in highly dynamic scenarios showed an increased number of false positives due to node mobility. These results suggest that an adaption of the threshold values might be required to lower the number of false positives. In order to gain further confidence in the results, additional evaluation has to be performed, e.g., GloMoSim simulations using the random waypoint mobility model.

Furthermore, we want to support additional routing protocols like AODV, which requires anomaly detectors that are tailored to the specific protocol. Additionally, our system currently only detects that an attack is running and can inform the user about that, but it is not yet aware of the position of the attacker within the network. Finally, an enhancement of our IDS to an intrusion prevention system (IPS) which arranges countermeasures should be accomplished.

Appendix

GlomoSim simulation settings	
GlomoSim version	2.02
Simulation time	600 s
Terrain dimensions	(1000, 1000)
Number of nodes	100
Node placement	GRID
Grid unit	100
Mobility	none
Propagation limit	-111.0
Propagation pathloss	FREE-SPACE
Noise figure	10.0
Temparature	290.0 K
Radio type	RADIO-ACCNOISE
Radio frequency	$2.4 \cdot 10^9$ Hz
Radio bandwidth	$2.0 \cdot 10^6$ bit/s
Radio rx type	SNR-BOUNDED
Radio rx SNR threshold	10.0
Radio tx power	7.87395 dBm
Radio antenna gain	0.0 dB
Radio rx sensitivity	-81.0 dBm
Protocols	802.11 / IP / OLSR

Acknowledgment

The authors like to thank Dr. Kilian Weniger and Ingmar Baumgart for providing the source code of PACMAN, and Dr. Erik-Oliver Blaß, Dr. Christoph Sorge, and Christoph Werle for valuable input.

References

[1] The OLSR daemon, version 0.4.10, January 2006.

[2] C. Adjih, D. Raffo, and P. Muhlethaler. Attacks against OLSR: Distributed Key Management for Security. In *OLSR Interop and Workshop*, pages 28–29, 2005.

[3] J. Cabrera, C. Gutierrez, and R. Mehra. Infrastructures and Algorithms for Distributed Anomaly-Based Intrusion Detection in Mobile Ad-hoc Networks. In *Proceedings of the IEEE Military Communications Conference MILCOM 2005*, volume 3, pages 1831–1837, 2005.

[4] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626 (Experimental), Oct. 2003.

[5] D. Dhillon, J. Zhu, J. Richards, and T. Randhawa. Implementation & Evaluation of an IDS to safeguard OLSR Integrity in MANETs. In *Proceedings of the International Conference on Wireless Communications and Mobile Computing IWCMC '06*, pages 45–50, New York, USA, 2006.

[6] A. Fourati and K. Agha. A Shared Secret-based Algorithm for Securing the OLSR Routing Protocol. *Telecommunication Systems*, 31(2):213–226, 2006.

[7] A. Hafslund, A. Tønnesen, R. Rotvik, J. Andersson, and Ø. Kure. Secure Extension to the OLSR protocol. In *Proceedings of the OLSR Interop and Workshop, San Diego*, 2004.

[8] F. Hong, L. Hong, and C. Fu. Secure OLSR. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications AINA 2005*, volume 1, pages 713–718 vol.1, 2005.

[9] J. Orset and A. Cavalli. A Security Model for OLSR MANET Protocol. In *Proceedings of the 7th International Conference on Mobile Data Management MDM 2006*, pages 122–122, 2006.

[10] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental), July 2003.

[11] D. Raffo, C. Adjih, T. Clausen, and P. Mühlethaler. An Advanced Signature System for OLSR. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 10–16, New York, NY, USA, 2004.

[12] M. Wang, L. Lamont, P. Mason, and M. Gorlatova. An Effective Intrusion Detection approach for OLSR MANET protocol. In *Proceedings of the 1st IEEE ICNP Workshop on Secure Network Protocols (NPSec)*, pages 55–60, 2005.

[13] K. Weniger. PACMAN: Passive Autoconfiguration for Mobile Ad hoc Networks. *IEEE Journal on Selected Areas in Communications*, 23(3):507–519, 2005.

[14] X. Zeng, X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a library for parallel simulation of large-scale wireless networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulation (PADS 98)*, pages 154–161, 1998.