Methodologies and Frameworks for Testing IDS in Adhoc Networks

Marko Jahnke, Jens Toelle,
Alexander Finkenbrink,
Alexander Wenzel
Research Institute for Communication,
Information Processing and Ergonomics
(FGAN-FKIE)
Neuenahrer Str. 20
D-53343 Wachtberg, Germany
{jahnke|toelle|finkenbrink|wenzel}
@fgan.de

Elmar Gerhards-Padilla,
Nils Aschenbruck, Peter Martini
University of Bonn
Institute for Computer Science IV
Roemerstr. 164
D-53117 Bonn, Germany
{padilla|aschenbruck|martini}
@cs.uni-bonn.de

ABSTRACT

Testing Intrusion Detection Systems (IDS) has been a substantial part of the development lifecycle, since the first prototypes and products appeared on the market. Unfortunately, many of the existing principles, procedures and systematic frameworks for testing IDS are not broad enough to cover systems which are focussed on mobile adhoc networks (MANETs).

As a baseline, this paper expands the most important requirements for IDS testing to MANET environments. Two alternative testbed realization approaches are described, including a common example scenario for comparing the properties of the approaches. One approach is based on hardware nodes, reproducible physical motion and radio signal attenuation; the other uses both hardware and virtual nodes and a motion emulation framework that is able to incorporate arbitrary radio propagation models.

A selection of MANET specific attacks and their implementation and impact on both types of testbeds is presented. These attacks are beyond the threats that we know from conventional wired networks, which still need to be taken care of in MANETs. Finally, the advantages of both testbed approaches are discussed. As a conclusion, a deployment strategy for testing MANET IDS under different conditions is derived.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Wireless communication

General Terms

Measurement

Q2SWinet'07, October 22, 2007, Chania, Crete Island, Greece.

Keywords

Hardware Testbed, Semi-Virtual Testbed, Attacks and Attack Implementations, Black Hole, Worm Hole, Smart MANET Jamming, reproducible node motion

1. INTRODUCTION

Testing intrusion detection systems (IDS), approaches and products has been a very active area of work since the mid-1990's. Several more or less representative testbed setups, cooperative experiments as well as reference network traffic datasets have been elaborated and published. In many deployment contexts, according systems and approaches have been evaluated, and articles about IDS product comparisons appear frequently in journals and on websites. However, several weaknesses in testing procedures, implementations and the interpretation of their results have been criticized in the past.

Unfortunately, most of the testbed setups and procedures are only suitable for wired networks or for wireless networks in infrastructure (managed) mode. The variety of impacts that a wireless mobile adhoc network (MANET) has on security – even if it still is an IP based network – is huge, as pointed out in many publications over the past few years. Although the quality of numerous network attack detection and protection approaches has been examined in various ways, a general guideline for testing and validating in these environments does not exist.

This article first summarizes the most important goals and requirements of IDS testbeds in infrastructure based network environments. After that, these requirements are expanded to MANET environments. The main part of this paper comprises the description of two alternative testbed realization approaches which are suitable for examining different aspects of the network behavior. A simple example setup (taken from a military IDS deployment scenario), and its implementation in both environments is described. After that, the ways of implementing general and MANET specific attacks in the testbeds are presented, followed by a discussion of the properties of the approaches. Finally, a conclusion is drawn on where to use which approach for obtaining best attack testing results.

2. IDS TESTBEDS: PURPOSE AND REQUIREMENTS

Testing hardware and software components for IT security is a basic part of the development cycle in order to ensure the quality of approaches, products and systems. The main goal of testing is to measure quantitative or qualitative properties of a test candidate during its operation. For comparing these properties, according metrics are needed.

When focusing on testing IDS, the most important properties to be determined are *detection capabilities* which include the question if a launched attack against one of the systems to be protected is detected by the candidate at all. In many case studies (e.g. [14], [9]), different IDS have been deployed in identical scenarios, and for every applied attack, it is determined whether the attack has been indicated by the candidate or not[1].

Additional questions to be answered by IDS attack tests include the following:

- How much time did the detection take itself ?
- How many resources did the detection require? What are the limiting factors of the detection under heavy load?
- How robust is the detection under stress and heavy load?
- How much useful background information is delivered (in order to perform a manual, semi-automatic or automatic attack response) ?

Conceptual insufficiencies in many test campaigns have been pointed out in the past, such as inaccurate assumptions on the representativeness of the background traffic and the way of presenting and interpreting test results (see [3, 12]).

An environment for applying a test series is called *testbed*. For IDS testbeds the following basic requirements can be derived:

1. *Representative architectures and platforms*
   The testbed network architecture as well as the node platforms (hardware, operating systems) need to be representative for systems to be protected by the candidate.

2. *Realistic services, applications and their dependencies*
   The running local and networking hardware and software services, as well as their inter-dependencies need to be representative for systems to be protected by the candidate.

3. *Realistic traffic / user modeling*
   The system and network usage need to be realistic. When focusing on effects on the network, realistic network traffic needs to be reproducibly generated (see e.g. [1]). When effects on hosts are examined, user and application behavior needs to be emulated (see e.g. [6]).

4. *Scalability and flexibility*
   Whenever a candidate is developed to deal with scalable and dynamic environments (e.g. in networks with hundreds of nodes and with dynamically changing members), the testbed setup needs to be dynamically changeable with a reasonable effort.

5. *Representative and reproducible attacks*
   The attacks applied to the testbed need to be exactly reproducible for every test run in order to get comparable results. Additionally, they need to be representative for a later deployment scenario, i.e. all potential attack varieties need to be mapped on the testbed.

Especially the last two requirements get unhandy very quickly when focusing on MANETs, as the next section turns out.

[1]This is obviously a non-precise result determination, since the question whether an attack indication really corresponds to the indicated attack is not trivial, especially if other simultaneous effects on the system (e.g. other attacks or malfunctionings) cannot be precluded.

3. SPECIFIC REQUIREMENTS FOR MANET IDS TESTBEDS

The requirements mentioned above also hold for wireless and MANET testing scenarios, since MANETs are also basically IP based networks. Any kind of network and host-based IDS from infrastructure based networks may be deployed in these scenarios and thus may need to be tested in these kinds of environments.

But in certain aspects, the testing requirements need to be extended. Especially the representativeness and reproducibility of the behavior of services, applications and users gets much more difficult than in infrastructure based networks. Several publications refer to testing network behavior in MANETs (e.g. effectiveness and reliability of protocols, [2], [15]). A few discuss the effects on testing IDS and detection approaches.

Patwardhan et al. [13] have developed a proof-of-concept implementation of a secure routing protocol as well as a routing independent intrusion detection and response approach for a MANET. They have validated their ideas using different blackhole and packet mangling attacks as well as layer 2 denial-of-service (DoS).

Karygiannis and Antonakakis [10] refer to their testbed framework that aims at keeping the correspondence between desktop simulations and outdoor field tests. The authors emphasize the increased benefit of their approach for testing IDS in a realistic environment.

The most notable difference between a MANET and infrastructure based networks is the mobility of the nodes, resulting in established and broken links which lead to a dynamically changing ISO/OSI layer 2 topology. So an additional aspect of user behavior in MANETs is the motion of the nodes that is realistic for the selected test scenario:

6. *Reproducible mode motion*
   The reproducibility of service, application and user behavior as mentioned in the previous section, does include the motion of every node. That means that the position, moving direction and speed need to be exactly reproducible to obtain comparable results.

   (a) *Effects on network connectivity, delays etc.*
   All potential effects of the node motion for behavior of the network need to be exactly reproduced. This includes parameters such as connectivity (circumstances of establishing and shutting down links), reliability (rates of successfully received packets, congestion effects etc.), and per-

formance measures (packet delays, retransmission rates, latency behavior).

(b) *Effects on services and applications*
Not only the structure and the traffic on the network are influenced by moving nodes. Also the behavior of applications and services that depend on geographic information (coordinates, relative positions, speed etc.) need to be reproduced, since they indirectly influence the behavior of the network as well.

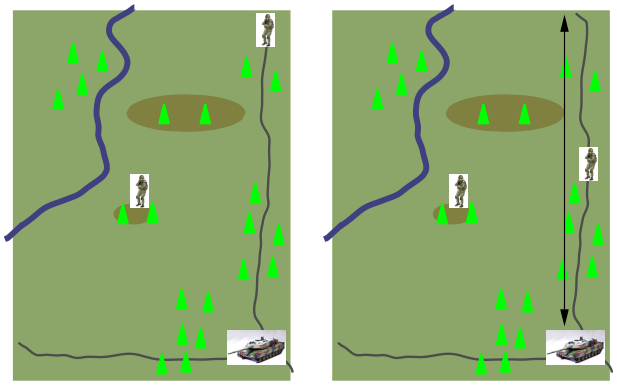
7. *Reproducible and representative MANET specific attacks*
Common attacks on wireless networks do not only include conventional network or host-based attacks, but also misbehavior on the physical and link layer (e.g. smart jamming attacks, outsider wormholes). These must be reproducible for every test case.

The necessary properties of the motion sequence strongly depend on the purpose of the systems that are to be protected by the candidate IDS. In this paper, we focus on military applications where MANETs support the infantry units in an operational area without any fixed communication infrastructure.

4. A SIMPLE TEST SCENARIO

Before describing and discussing alternative approaches for testing IDS in MANETs, a common test scenario is necessary for reference. The testing activities we consider aim at determining whether a distributed set of wireless network sensors is able to detect malicious routing manipulations and packet drops.

As an example scenario, we consider a military infantry reconnaissance mission, as depicted in Fig. 1. In the static case (Fig. 1(a)), an infantry unit reconnoitres the area (node in the north-east) and communicates with a command node staying in the background (node in the south-east). Another infantry unit (node in the center) secures the area and the reconnoitring unit. In the mobile case, the reconnoitring



(a) Static (b) Mobile

Figure 1: Two instances of the example scenario

unit moves continuously between the lower and the upper edge of the scenario plane (Fig. 1(b)).

Each of the units operates a mobile wireless network node, such as a high performance PDA or an Ultra-Mobile PC with according network interfaces, supporting the fighters with VoIP communication and command&control systems including navigation and tactical information. The network characteristics are based on IEEE 802.11b.

In the following sections, two alternative implementations of this scenario are described and discussed.

5. HARDWARE TESTBEDS AND PHYSICAL MOTION

In this section, we describe our first approach for a realistic *Hardware Testbed Environment*. We use real hardware and physically motion to fulfill parts of the requirements specified in Sect. 2 and 3. First, we describe our setup including the used communication hardware, the physical motion, and the environment we use the testbed in. After that, some results of measurements of relevant network behavior parameters are presented.

5.1 Setup

As specified in Sect. 2, it is necessary to use representative architectures and platforms. In a typical MANET, mobile devices like laptops, handhelds or PDAs are used. In the current setup, we have chosen Sharp Zaurus SL C3100 and Sharp Zaurus SL 5500 devices.

One central challenge of a testbed using hardware is the realization of reproducible node motion. In recent related work (e.g. [8], [7], [11]), human beings carry the communication devices. They have been told to move on a fixed route with constant speed. This does not allow exact measurements or identical replications because there is no way to assure that the test persons really move according to the specification. In order to realize this, we figured out four requirements a medium moving the nodes has to meet:

- *Fixed route*
It must be possible to specify a fixed route for the movement and to do multiple replications using exactly the same route.

- *Constant speed*
It is necessary to enable the user to specify constant speed. Therefore, the medium must be able to move with exactly constant speed.

- *Adequate ground*
For the test setup adequate ground (e.g. solid, flat) is required. The ground has to enable the medium to travel the fixed route with constant speed on arbitrary replications.

- *Transport of communication hardware*
The medium has to be big enough to transport the communication hardware.

A model railway fulfills the above requirements for a medium moving the testbed nodes. There are different track gauges for model railways. In order to be able to transport the used handhelds we choose the German gauge G which has a track width of 45 millimeters. The speed of the model railway depends on the connected voltage. Therefore, an external transformer is used to obtain constant voltage for ensuring a constant speed of the model railway.

For realistic measurements in MANETs, it is not only necessary to consider one-hop but also to assure multi-hop

connections. There are two ways to assure multi-hop connections: use large space and normal range of the nodes or use small space and decreased range. Measurements in changing environments lead to non-reproducible effects. But the bigger the space used, the more difficult it is to provide a consistent environment. Thus, we use the second possibility in order to minimize the space needed. To minimize effects of the environment on the measurements we use one dedicated room for the measurements.

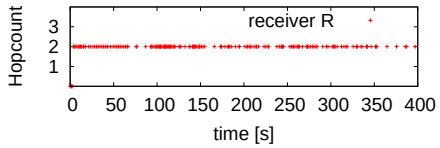
5.2 Network Behavior

The testbed was configured as a downscaled instance of the selected scenario as described in Sect. 4. Static nodes have been placed at the according locations in the testbed area, and the moving node was placed on the railway; the tracks have been placed accordingly on the eastern boundary of the testbed area. During the test run, the metrics *Hop Count* and *packet delivery fraction* have been determined.

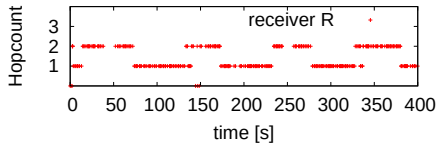
First, we show measurements regarding whether it is possible to assure multi-hop connections with our approach. This is essential for a MANET testbed. After that, we introduce measurements of the packet delivery fraction in our testbed. These show the impact node mobility has on our system.

5.2.1 Hop Count

Fig. 2 presents the Hop Count (number of nodes which are participating in the transport process of IP packets from sender to receiver) for the static example scenario and the mobile example scenario in the hardware testbed. The goal was to show that there are multi-hop connections in our testbed. For determining this parameter, we look up the number of hops to the receiver in the routing table of the sender every time the routing table potentially changes.



(a) Static

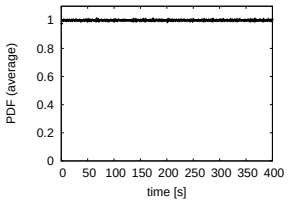


(b) Mobile

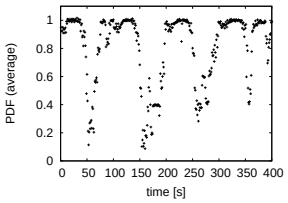
Figure 2: Hop Counts in hardware testbed example scenario

Fig. 2(a) shows the Hop Count of the route between reconnoitring and command node (values of zero represent lost packets). The Hop Count stays at two for the whole measured time, since the route leads through the securing node. Therefore, with our approach it is possible to assure multi-hop connections.

The geographic setup for the measurements in the mobile scenario is the same as for the ones in the static scenario. This time, we have periods with one-hop and periods with two-hop connections. If the reconnoitring node is in direct communication range to the command node, they communicate over one hop. Otherwise, traffic is routed over the securing node and therefore, the route has a length of two hops.



(a) Static



(b) Mobile

Figure 3: PDF in hardware testbed example scenario without attacker

5.2.2 Packet Delivery Fraction

In order to show the impact of mobile nodes on the network, we utilize the packet delivery fraction (PDF). We calculate the packet delivery fraction for our static and mobile example scenario. At this point, the goal is to show the impact of mobility on our system. We consider VoIP communication between the reconnoitring node and the command node. For each version of our scenario, we perform ten replications. There are big variations between the different single replications but due to clarity of the graphics we do not present confidence intervals. In reality these variations are typical since there are several reasons for such variations (e.g. interfering nodes or obstacles moving in the communication area). In each single replication, we calculate the PDF over intervals of one second.

Fig. 3(a) shows the PDF of the static version of our scenario. As expected, the PDF stays at value 1.0 for the whole duration of the measurements. Fig. 3(b) comprises the PDF of the mobile scenario. At the beginning, the PDF is as high as in the static case. The variations of the PDF are due to the moving reconnoitring node. The moving node leads to degradation of the signal strength and stale routes in the routing tables of the nodes. Furthermore, there are some massive obstacles along the way of the moving node influencing the PDF.

Note that there are variations of the PDF visible in Fig. 3. These variations are due to variations of signal strength which occur in reality and can have several reasons.

6. SEMI-VIRTUAL TESTBED WITH MOTION EMULATION

A completely different approach for reproducible node motion in MANETs comprises a setup of both real and virtual nodes. This is called the *Semi-Virtual Testbed Environment*. It also fulfills parts of the requirements specified in Sect. 2 and 3.

6.1 Setup

The semi-virtual testbed consists of the following components which are altogether connected using real radio network adapters:

- *Hardware Nodes*
  At least one hardware based node exists in the testbed.

It comprises a representative mobile hardware and operating system platform as well as network services and applications which are connected to other nodes via the built-in radio network adapter using the standard OS interface access.

- *Virtual Nodes*
  A number of virtual nodes may also be deployed in the testbed, i.e. software compartments running separate instances of network services and applications (virtual nodes) on a single hardware node (virtual host). The access from the virtual nodes to the radio network is implemented using so-called bridge interfaces. Note that a real radio communication between virtual and hardware nodes does only work if the medium access addresses of the virtual nodes are mapped on the radio adapter, i.e. they are visible on the medium.

The virtual nodes may be realized using complete hardware emulation or other virtualization techniques. In our current experimental setup, we use OpenVZ[2] to allow sharing of OS resources. Techniques for user and application behavior modeling can be deployed on both virtual and hardware nodes. In our lab, we use Sharp Zaurus 3000CL running OpenZaurus, HP iPAQ running Familiar Linux, FlyBook subnotebooks and laptop computers running Debian GNU/Linux. The two latter are capable to serve as virtual hosts.

The more difficult part of the testbed is the motion of the nodes. In contrast to the hardware testbed where the reproduction of motion is based on physically moving hardware, a different approach for covering both hardware and virtual nodes is needed.

As mentioned in different publications (e.g. [4], [2, 15]), blocking interfaces using kernel level mechanisms is a technique which can easily be implemented in a reproducible manner. But in contrast to a simple scripted on/off switching of connections (e.g. using IPtables[3]) we suggest a more general approach that incorporates different external influences, including node motion, radio propagation model and potential obstacles. This framework is called the *MotionEmulator*. The basic architecture of the semi-virtual testbed and the MotionEmulator's principles of operation are depicted in Fig. 4. The MotionEmulator operates in different phases:

1. *Distribution and Initialization*
   During the first phase, the geographic information (i.e. positions, moving directions, and speed) about all nodes during the later emulation run is available from a database on the emulation server (*MotionServer*). Every node in the semi-virtual testbed also contains a component called *MotionClient* that initially connects to the server and requests this information about **all** nodes in the network. After all clients have received the data, a synchronous start signal is given to all clients, and they disconnect immediately.

2. *Emulation Run*
   After the start signal, every client runs a continuous loop through its local geographic database and primarily determines the coordinates of the local node. These coordinates are fed to an interface of the local GPS (Global Positioning System) information server (daemon), that is usually receiving data from a dedicated hardware. During the emulation run, all local services and applications which depend on the GPS daemon are operating on the emulated data.

   Additionally, a second component on every node in the testbed calculates continuously the relative positions to all other nodes in the network. Based on the deployed radio propagation model, it determines the quality of the **logical** connection between the local node and the others. If not every packet should be received from a certain node (e.g. due to a large geographic distance, disturbances in the radio propagation model or obstacles in the line-of-sight), the current percentage of packet loss within a given time period (e.g. 1s) is determined. In our emulation, this percentage of packets is then dropped randomly at the sender side and thus cannot reach the receiver.

[2] `http://www.openvz.org`
[3] `http://www.linux.netfilter.org`

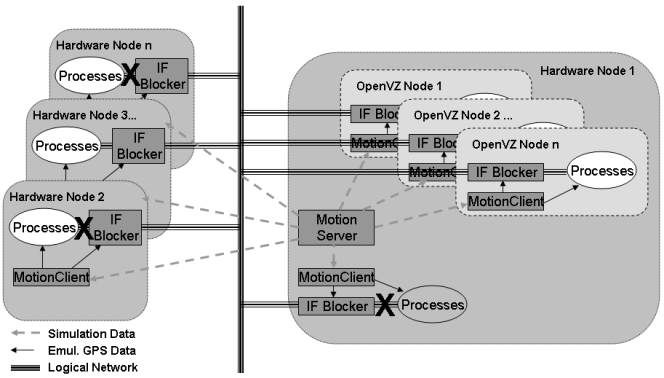
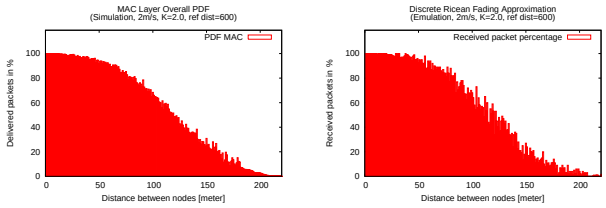


Figure 4: Operating principles of the *MotionEmulator* in the previously shown semi-virtual testbed setup.



(a) Simulation PDF (b) Received Packets

Figure 5: Resulting packet delivery fraction of the propagation model in a simulation run and the actually received packet count from the semi-virtual testbed approximation

Basically, an arbitrary theoretical radio propagation model can be applied in order to determine whether a radio packet is delivered to its receiver or not[4]. As an example, we have implemented a log-normal distribution of the signal strength to approximate radio wave propagation in free space without obstacles. Currently, more complex models, such as the

[4] The reliability and potential complexity of radio propagation models are under continuous discussion and beyond the scope of this paper.

the generic model from the BoMoNet suite[5] is integrated. The resulting PDF of one simulation run of the model is depicted in Fig. 5(a). For the approximation, an average percentage of packet loss for distance steps of 1 meter has been determined over several runs. This percentage is used as a parameter for the IPTrandom module which then actually drops the packets at the sender during an MotionEmulator run. Fig. 5(b) shows the measured percentage of received packets in the semi-virtual testbed. The remaining discrepancy of the two curves is due to the usage of integer calculations for the approximation.

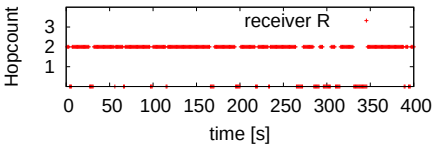
6.2 Network Behavior

To examine the usability of results obtained in the semi-virtual testbed, it is necessary to recreate the behavior of the hardware based testbed using the same example scenario as described in Sect. 4. The goal was to obtain comparable network behavior results, especially the multi-hop characteristics and the changing topology.

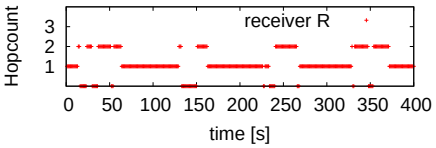
The first step to achieve this was to obtain the geometric properties of the hardware testbed setup and to build a motion sequence, that maps the positions, moving directions and the speed of the hardware nodes. The results of the application of this motion sequence are as follows:

6.2.1 Hop Count

Fig. 6 presents the Hop Count for the static example scenario and the mobile example scenario in the semi-virtual testbed. The goal was to show that the results are at least comparable to the measurements in the hardware testbed, although real-world effects on the radio propagation (e.g. attenuation, reflexion, fraction) are not emulated. Basically, the Hop Counts stays at a value of one; the relatively high number of lost packets (Hop Count at zero) compared to the hardware setup is an effect of the random influences in the fading model.



(a) Static

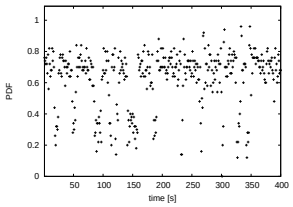


(b) Mobile

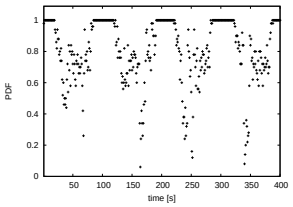
Figure 6: Hop Counts in semi-virtual testbed example scenario

6.2.2 Packet Delivery Fraction

Fig. 7 presents the according PDF values for the semi-virtual testbed. In contrast to Fig. 3(a), where we have an almost constant fraction of one, Fig. 7(a) shows a more scattered picture. This is due to the deployment of the Ricean fading model, where some packets are expected to get lost in the assumed distance from sender to receiver. In the hardware testbed, almost all packets sent are received, due to propitious reflexion effects.



(a) Static



(b) Mobile

Figure 7: PDF in semi-virtual testbed example scenario without attacker

The general behavior of the PDF in Fig. 3(b) and 7(b) is comparable, though not equal. On one hand, this is again due to a different radio propagation. On the other, average values are depicted in Fig. 3(b), whereas Fig. 7(b) presents one single measure result[6].

7. ATTACKS AND ATTACK IMPLEMENTATIONS

In order to analyze the feasibility of a new intrusion detection approach for MANETs both the mobility of the network nodes and the potential attacks have to be considered. This section discusses the implementation and some of the impacts of MANET specific attacks in both testbed variations; conventional attacks – those which may also occur in wired networks – are not in the focus of this section.

7.1 Blackhole Attack and Variations

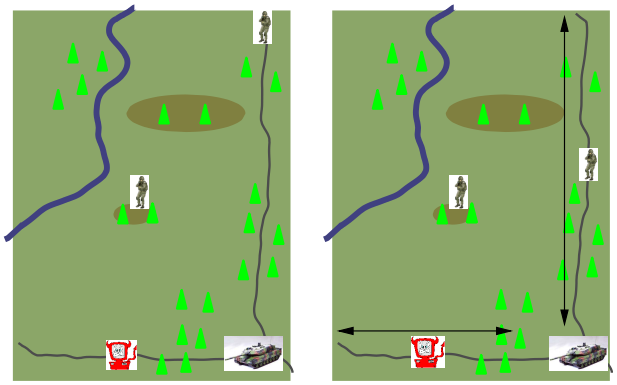
The routing attack we consider first is called a *blackhole* attack. This is an attack that can be performed relatively easy by network insiders (legitimate MANET nodes which already participate in the routing and forwarding process). The goal of a pure blackhole attack is to become part of as many routes of the network as possible, and (generally or selectively) drop packets in order to perform a DoS attack against the other nodes.

The first step can be achieved by sending fake routing information to the other host, pretending that the attacker node is very close to MANET nodes which are far away. So when establishing a route to these far away nodes, a neighbor node will choose the attacker as the next hop. The second step for the attacker is just to drop the packets to be forwarded, either all of them, or selectively (to suppress certain network services, or just to make the attack harder to identify). Further on, this attack or variations thereof may be used as a stepping-stone for other intrusive procedures.

For determining the network behavior in presence of this kinds of attacks, we modified our previously described scenarios by introducing an additional attacker node, as depicted in Fig. 8(a) and 8(b).

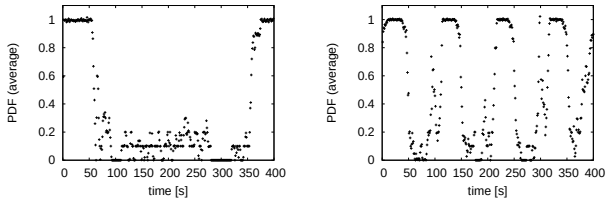
[5] `http://web.informatik.uni-bonn.de/IV/bomonet/ns2.htm`

[6] Average values do not make sense in the semi-virtual testbed, since no differences in the results are expected as long as the instance of the fading model has not been recreated.



(a) Static (b) Mobile

Figure 8: Example scenarios with blackhole attacker node



(a) Static (b) Mobile

Figure 9: PDF in hardware testbed example scenario with blackhole attacker

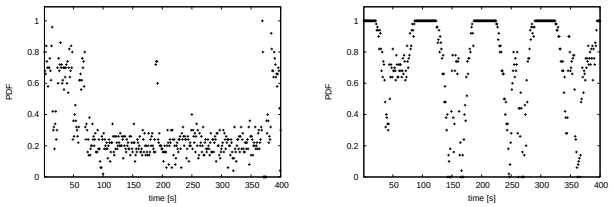
In the following, we introduce measurements of the packet delivery fraction in our testbeds. These show whether our blackhole attack emulation works and provides an indication of the impact on our systems.

7.1.1 Hardware Testbed

Using the modified scenarios with an attacker, the measurements were performed in the same way and with the same parameters as described in Sect. 5.2.2. The faked routing messages are generated by a patched instance of the routing daemon. After becoming part of as many routes as possible, an IPtables based script drops all packets which are to be forwarded.

In Fig. 9(a), the PDF of the static scenario with attacker is presented. The attack starts at second 60 and ends at second 360. Compared to Fig. 3(a), the PDF drops significantly but stays above zero. This is due to the use of the ETX metric (see [5]) in the routing protocol which leads to the fact that the blackhole is not completely succeeding in capturing the route. After the attack ends the PDF raises back to one again.

Fig. 9(b) introduces the PDF of the mobile scenario with attacker. The attack again starts at second 60 and ends at second 360. At the beginning of the measurements the reconnoitring node is in direct communication range with the command node. Thus, the attacker does not affect the connection (PDF is at one). When the reconnoitring node moves out of direct communication range the blackhole captures the route and the PDF drops to zero. This behavior can be seen four times in Fig. 9(b). Compared to Fig. 3(b), in this scenario the PDF drops to smaller values and stays



(a) Static (b) Mobile

Figure 10: PDF in semi-virtual testbed example scenario with blackhole attacker

at a value significantly below one for longer periods. The impact of the attack is clearly visible in both, the static and mobile scenario.

7.1.2 Semi-Virtual Testbed

The implementation of the blackhole attack in the semi-virtual testbed does not differ from the hardware environment, with the notable exception that the false routing messages are generated by a script that crafts according packets on the IP layer and injects them into the IP stack. Depending on the interface driver, this approach does not require further modification of the system software.

The resulting PDF from Fig. 10 shows comparable differences to those obtained in Fig. 9 with the same reasons as explained in Sect. 6.2.2.

7.2 Smart MANET Jamming

Another example for a MANET-specific attack is a layer 2 DoS attack using another wireless network adapter as a jammer. In contrast to a broadband jammer, disturbing effects on layer 2 may not be easily distinguishable from external influences on the radio propagation, malfunctionings or misconfigurations, and thus may be hard to detect. Further on, these attacks may not need as many battery power as a constantly emitting broadband jammer.

7.2.1 Hardware Testbed

The impact of using a smart jammer can easily be analyzed in a hardware testbed. In order to obfuscate the attack it is reasonable to generate a situation similar to the hidden terminal problem. To create a hidden terminal, in a hardware testbed it is sufficient to use a setup like the one depicted in Fig. 11. We use ellipses to illustrate transmission ranges although we know that this is not realistic. Nodes N1 and N2 use low transmission power leading to transmission ranges shown by the black ellipses. The nodes are in transmission range of each other, but do not disturb the transmission and carrier sensing of nodes A1 and A2. The attacker nodes A1 and A2 use high transmission power. Their transmission ranges are shown by the red ellipses. Node A1 constantly transmits data to node A2. Therefore, the carrier sensing of nodes N1 and N2 shows a busy medium and these nodes are not allowed to transmit data.

7.2.2 Semi-Virtual Testbed

The integration of a jammer into the semi-virtual testbed needs more complex considerations. As an example, the radio propagation model used in the emulator has to calculate the signal strengths of the adjacent nodes and of the jammer depending on the current (simulated) positions of all

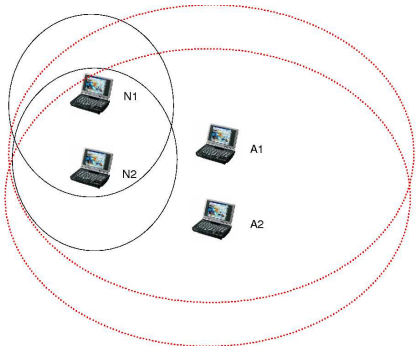


Figure 11: Jamming in Hardware Testbed

the MANET nodes and the jammer. Based on this values, a probability needs to be calculated for each MANET node, whether the reception of MANET traffic is possible or not. This calculation has to be updated at least every time one of the nodes or the jammer changes its position. These costs increase the complexity of the emulation. Due to this and to the expected limited applicability of the results, further investigations have been omitted.

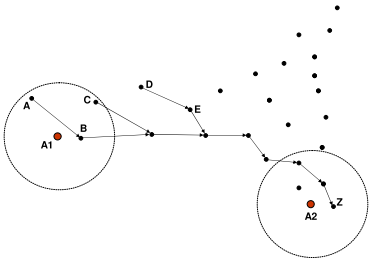
7.3 Wormhole effects in complex MANET testbed setups

A more complex attack that has several interesting effects on MANETs with more than just a handful of nodes is the *outsider wormhole* attack. This attack aims at creating a "shortcut" in the routing topology and allows – amongst other attack possibilities – separation of network segments – without being a legitimate part of the network. The basic idea of this attack is as follows:

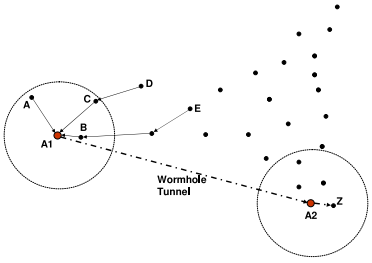
1. Two cooperating network nodes are equipped with an out-of-band link (e.g. VHF, UHF) that works over larger distances than the MANET radio.
2. The attacker nodes are placed at different locations in the network, having no direct connection to each other over the MANET radio.
3. Each attacker (passively) receives all traffic in its MANET radio range and passes it to the other node over the out-of-band link. The other attacker immediately replays the received traffic on his radio adapter (including the link layer address of the original sender).

This easily induces not only a twisted topology to the network (that may have many effects which are not easily determinable). It also may have the effect that routes from large parts of the network are attracted and thus dropping packets may damage the network seriously, without being able to easily detect the source of the attack. An example for the routing behavior in such a situation is depicted in Fig. 12.

Generally, this attack is only implementable, as long as it can be ensured that the wormhole endpoints (attacker nodes) cannot receive the replayed packets from each other over the MANET radio. If they could, this feedback loop would induce a constantly raising amount of repeated packets that will break down the network sooner or later.



(a) Inactive wormhole



(b) Active wormhole between Attackers A1 and A2

Figure 12: Routing from in presence of an outsider wormhole

7.3.1 Hardware Testbed

Generally, this attack seems to be implementable in a hardware based testbed, but when using attenuated signals, it becomes very difficult to avoid that one wormhole endpoint receives packets from the other. Due to this and to enormous efforts necessary for establishing a reproducible setup with more than 8 nodes, further investigations have been omitted.

7.3.2 Semi-Virtual Testbed

There are two possibilities to implement this attack in the semi-virtual testbed:

- *Imitate the hardware based implementation*
  Imitating the hardware based attack implementation is a non-trivial task. This is mainly due to the previously mentioned possibility to receive each others repeated traffic that would lead to a feedback cycle. In our semi-virtual testbed, we have created an according setup: We are using two dedicated attacker nodes which are additionally connected via an ethernet link that is not visible to the rest of the network. On the attacker nodes, the radio interfaces are switched to promiscuous mode, and all radio traffic is sniffed using the standard tcpdump[7] tool. The sniffed traffic is passed through a pipe over the ethernet link and then is fed into a tcpreplay[4] process that is able to inject the packets into the radio device which immediately replays them on air.

[7] `http://www.tcpdump.org`

To avoid the mentioned feedback cycle, different mechanisms may be used, such as artificial VLAN tagging of the MAC layer frames. Additionally, effects of the virtual interfaces need to be taken into account, such as layer 2 messages (e.g. ARP requests) which are delivered directly via the virtual interface rather than being sent over the air.

- *Implement the attack effects*
  To implement just the effects of the attack in the testbed setup is a lot easier to achieve in the semi-virtual testbed. The blocked connection between two nodes which are too far away from each other to receive MANET traffic directly, just needs to be unblocked for the time the worm hole operates. Delays and other effects of additional wireless links can be emulated through IPtables modules, potentially transporting the packets to the user space to be processed and back again. Using this approach, the attacker nodes themselves do not need to be instantiated in the testbed.

Effects of the out-of-band link (e.g. delays, packet loss) can also be emulated using IPtables based mechanisms, as our first experiments have shown.

8. DISCUSSION

The approaches to evaluate intrusion detection systems in MANET environments with dynamic network node behavior presented in the Sect. 5 and 6 show both advantages and disadvantages. This section discusses them and analyzes the appropriate evaluation method subject to the question under investigation.

Both approaches have the useful property that almost real system platforms (OS, applications, user interfaces) are deployed. Thus, the effort for installing, configuring, and cloning realistic services and applications – including the candidate IDS itself – is relatively low, compared to e.g. a pure simulation setup in a discrete event based network simulator. This also holds for the way attacks are implemented. As long as they do not require special hardware, they can be realized straightforward on nodes in the testbeds with a relative low effort. Additionally, usage of real hardware nodes has the additional benefit of being able to demonstrate system properties to potential customers. This is often much more descriptive than presenting theoretical results from software simulators.

Whenever physical effects like radio performance may have a direct or indirect impact on the candidate IDS or any of its components, real network hardware needs to be used, e.g. network interface cards and antennas with directional radio pattern. This can only be carried out in a hardware based testbed, since the physical world behavior of these system components is difficult to model in emulation based setups.

In the hardware testbed, layer 2 effects can be studied in a real world environment. Whenever a performance analysis for an IDS in a MANET needs to be conducted, an accurate study of the radio properties of the environment where the MANET will operate is needed. If it is possible to operate a hardware testbed immediately in the later deployment location or in a comparable environment, a realistic analysis of radio influences is possible.

The usage of the real hardware allows the early detection of real world problems, e.g. discrepancies between network protocol standards and actual implementations. In addition, it offers the possibility to immediately analyze the influence of changes in the system setup, e.g. influences of obstacles in the area of the radio propagation. The consequences can be studied directly, researcher and interested parties can instantaneously see changes in the system behavior and in the system performance.

But when using hardware testbeds, the investigator needs to know about the following disadvantages. The installation of a hardware testbed is complex and both time and space consuming. Manual work has to be carried out, and the hardware testbed needs supervision and maintenance. Beside the fact that the testbed devices have to be supplied, the man-power needed to operate such a testbed might decrease the attractiveness of this approach due to the significantly higher costs.

The work to install a hardware testbed and the devices needed to move network nodes limits this approach to a small number of mobile nodes. Analyses with dozens or even hundreds of nodes do not appear to be reasonable or even practicable. Another limiting factor is the size of the available experimental area, since large areas drastically increase the costs and complicate the installation, the experiments and the maintenance. The size of the areas needed for the testbed setup can be decreased using the techniques presented in Sect. 5, but both using attenuated antennas and decreased sending power degrades the quality of the results of the experiments. The influence of these procedures and their significance for real-world application of the tested systems have to be analyzed and considered.

In contrast to this, the usage of software network and mobility emulation allows an easier setup of large-scale scenarios, but raises different challenges. Using an according framework for applying node motion and radio propagation information for emulating real-world effects in a validated manner (as the MotionEmulator described in Sect. 6) decreases the amount of effort to create new setups significantly. Only if scenario specific effects need to be implemented, additional challenges and efforts arise, such as the effects of an out-of-band link in the outsider wormhole attack, see 7.3.2.

In general, besides the amount of available resources (computing power, memory) there is no categorical limit concerning the size of the semi-virtual scenario. Scenarios with hundreds or thousands of mobile nodes are possible, but only reasonable as long as the investigator knows how to analyze and interpret the results.

9. CONCLUSION AND FUTURE DIRECTIONS

This paper has presented two alternative approaches for implementing IDS testbeds that fulfill most of the collected existing requirements – on one hand, a purely hardware based approach using attenuated radio signals and physical motion, and on the other a semi-virtual approach with both hardware and software nodes that emulates the motion and its effects on the network.

Both approaches have their own advantages but do not perform perfectly in all possible test cases. As a result, we recommend a combination of both approaches. Fundamental scenarios should be analyzed using both techniques. Whenever real-world effects (e.g. radio signal propagation) are in the focus of the examination, hardware based setups lead to most realistic results, whereas examination of larger scaled phenomena may be only feasible in the semi-virtual testbed. The selection of the environment should be according to the specific testing requirements and effort considerations as discussed in the last section.

Our current and future work includes the incorporation of more complex radio propagation models (maybe including obstacles) and the examination of more complex scenarios in order to develop more robust attack detection and prevention techniques for tactical MANETs.

10. REFERENCES

[1] S. Antonatos, K. Anagnostakis, and E. Markatos. Generating Realistic Workloads for Network Intrusion Detection Systems. In *Proc. of the 4th International Workshop on Software and Performance (WOSP'04)*, 2004.

[2] Ad hoc Protocol Evaluation testbed (APE) Project Documentation, 2002.

[3] S. Axelsson. The Base-Rate Fallacy and its implication for the Difficulty of Intrusion Detection. In *Proc. of the 6th ACM Conference on Computer and Communications Security*, 1999.

[4] W. Chao, J. Macker, and J. Weston. The NRL Mobile Network Emulator. Technical report, Naval Research Laboratory, 2003.

[5] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless networks. In *Proc. of the 9th annual Int. Conference on Mobile Computing and Networking*, 2003.

[6] H. Debar, M. Dacier, A. Wespi, and S. Lampart. An Experimentation Workbench for Intrusion Detection Systems. Technical report, IBM Research, 1998.

[7] R. Gray. Soldiers, Agents and Wireless Networks: A Report on a Military Application. In *Proc. of the 5th Int. Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'2000)*, 2000.

[8] R. Gray, D. Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan. Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms. Technical report, Dartmouth College, Computer Science, 2004.

[9] C. Iheagwara and A. Blyth. Evaluation of ID systems in a switched and distributed environment: the RealSecure case study. *Computer Networks*, 32.

[10] A. Karygiannis and E. Antonakakis. mLab: A Mobile Ad Hoc Network Test Bed. In *Proc. of 1st Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, Santorini, Greece, 2005.

[11] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordström, and C. Tschudin. A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations. In *Proc. of the IEEE Wireless Communications*, 2002.

[12] J. McHugh. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Transactions on Information and Systems Security*, 8, 2000.

[13] A. Patwardhan, J. Parker, A. Joshi, A. Karygiannis, and M. Iorga. Secure Routing and Intrusion Detection in Ad Hoc Networks. In *Proc. of the 3rd IEEE Int. Conference on Pervasive Computing and Communications*, Hawaii, 2005.

[14] K. Richards. Network based intrusion detection: A review of technologies. *Computers & Security*, 18, 1999.

[15] C. Riechmann. MANET Forwarding Protocol (MFP) for Multicast and Unicast Traffic. Technical report, Wachtberg, Germany, 2006.