Experiences From Security Research Using a Wireless Mesh Network Testbed

André Egners†, Patrick Herrmann†, Tobias Jarmuzek*†, and Ulrike Meyer†
Research Group IT Security, RWTH Aachen University, Germany
† {surname}@itsec.rwth-aachen.de
* {tobias.jarmuzek}@rwth-aachen.de

***Abstract*—Wireless Mesh Networks (WMN) consist of a wireless infrastructure of mesh routers which are connected to the Internet via mesh gateways. In recent years many testbeds for WMNs have been implemented to test and evaluate different aspects of WMNs, however, none of these has been designed with testing and evaluating security mechanisms for WMNs in mind. In this paper we share our experience with designing a testbed dedicated to testing and evaluating security protocols in a realistic setting. We detail the hardware and software setup of our testbed, the management tools we developed to facilitate maintenance of our testbed. Finally, we show the potential of our testbed by presenting experimental results we gained using our testbed.**

***Index Terms*—Testbed, Wireless, Mesh, Networking, Security.**

I. Introduction

Wireless connectivity using heterogeneous access technologies on various devices becomes more important each day. One of the most popular wireless access technology nowadays is the IEEE 802.11 standard. It is used in private homes, the public sector as well as in enterprises. In the latter two cases, an operator typically provides an infrastructure consisting of access points connected by wire to a wired backbone network. Providing a wired infrastructure is, however, a costly endeavor, needs careful planning, and often results in static inflexible structures. Wireless Mesh Networks (WMNs) aim to overcome these problems by a wireless infrastructure. In particular, WMNs consist of Mesh Routers (MRs) which are connected to the Internet via Mesh Gateways (MGs). MRs may also act as Network Access Server (NAS) to Mesh Clients (MCs). MCs connected to a WMN can communicate with other MCs on the same WMN or any other node on the Internet. In addition, MCs may also act as MRs.

In recent years many testbeds for WMNs have been implemented to test and evaluate different aspects of WMNs such as the behavior of routing protocols or the performance of TCP over multiple wireless hops [1], [2], [3], [4], [5]. None of these has, however, been designed with testing and evaluating security mechanisms for WMNs in mind. At the same time, security protocols have been proposed that aim at protecting WMNs [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], covering different aspects of WMN security. However, none of these has been evaluated in a real-world testbed, which makes it very hard to assess the practical use of these proposals let alone to fairly compare them with each other.

In this paper we contribute to closing this gap by sharing our experience with designing a testbed dedicated to testing and evaluating security protocols in a realistic setting. In particular, we detail the hardware and software setup of our testbed, describe the management tools we developed to facilitate maintenance of our testbed. Finally, to illustrate the potential of our testbed, we present experimental results we gained using our testbed. This includes a general evaluation of throughput, packet loss, and RTT as well as performance results gained while testing our own security and handover protocols.

II. ITSec Testbed

A. Requirements

As our research focus is wireless security, i.e., specifically security of WMNs, we decided that practical demonstration and feasibility studies by using a testbed are of significant importance to us and the research field. Therefore, we analyzed a variety of existing testbeds. We quickly realized that all of these have been created for specific purposes, e.g., researching routing algorithms, or increasing the performance of transport protocols. As such, all the testbeds have specific setup and a range of tools and functionalities which they offer the researchers working with the testbeds. For us, the *primary* requirements were the following: (1) We need to be able to fully control the network topology and the respective routing protocols. This has been proven to be important for researching handover protocols and testing them. For instance B.A.T.M.A.N. [19] now includes extensions to signal handover from one access point to the next. Other routing protocols without this feature would therefore negatively impact the performance of handover protocols. (2) Full control over the mode 802.11 WLAN is operating in, i.e., devices being in station and master mode as opposed to all nodes running in ad-hoc mode. This has significant advantages as the full potential of the 802.11i security mechanisms can be leveraged. Besides the primary requirements, we also derived *secondary* requirements for our testbed: Complete physical control over the nodes in an easy manner. Thus, sharing a testbed with other researchers across countries was not an attractive concept, especially when crashing the nodes. No scheduling of slots for using the testbed. This would otherwise create a lot of unnecessary overhead while limiting the pace of process, e.g., when evaluating the performance of network based mechanisms. In the past this has proven to be of great relevance, especially

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

for students working towards their Bachelor or Master theses. As other testbeds, using Linux on the nodes is superior to any other choice as it allows the necessary software to be written in most common programming languages.

Hardware components are also similar in all testbeds we analyzed. Most devices are equipped with Atheros chips for wireless connections and have sufficient RAM and calculation power. A manual installation of the operating systems on each node is obviously not efficient. A comfortable approach to flash multiple nodes with system images is a combination of a Trivial File Transfer Protocol (TFTP) [20] server and a DHCP [21] server. Before booting from the hard disk, the nodes try to get a DHCP lease from the Ethernet interface and load a small system image into its RAM from the TFTP server. This temporary system contains tools to mount a network share containing the desired node system image and to flash it onto the nodes' persistent storage. Nodes should easily be able to chose whether to boot from a hard disk or their network interface. After the process of flashing the node, some node settings must be changed to make it work with the testbed, i.e., setting IP addresses and routing information.

Hostapd [22] and wpa_supplicant [23] are the standard software for wireless connections and are used in all analyzed testbeds with infrastructure mode. Remotely managing configurations and controlling these daemons is a necessity and should be possible in a convenient way. It also has to be considered that the nodes may differ in hardware. Different hostapd and wpa_supplicant configuration files must be applicable to different interfaces on various nodes.

We also need to be able to manage the RADIUS server of hostapd as it is required for authentication of all nodes. As opposed to the de-facto standard RADIUS server, freeRADIUS [24], it does not provide a management interface, yet. Different nodes in a WMN testbed often have similar configurations and only differ in small details, e.g. wireless authentication credentials. We need to be able to provide node-profiles or similar functionality to assign settings to multiple nodes.

B. Architecture

The network consists of MRs, MCs, MGs, and a central management server (meshctrl). The mesh routers and clients are interconnected using infrastructure mode. By using the Extensible Authentication Protocol (EAP)-Tunneled Transport Layer Security (EAP-TTLS) [25], each connection is protected by Wi-Fi Protected Access 2 (WPA2) on the link layer. MGs are responsible for routing the traffic to other domains, e.g., the Internet, using a wired backbone. Since each node is authenticated based on EAP, an Authentication, Authorization and Accounting (AAA) Server is necessary. The *meshctrl* server is connected to the mesh network by wire and implements the RADIUS server included in hostapd to authenticate mesh routers, clients and gateways. Also, the management web interface of *pwrmesh* (cf. SectionII-E) is hosted on this server.

C. Hardware

All nodes run on the *PC Engines ALIX.3D3* boards which are equipped with a 500 MHz AMD Geode LX800 CPU, an on-chip 128 bit AES Security Block, and 256 MB DDR DRAM. Persistent storage is realized by using 16 GB Compact Flash cards which can be plugged into a CF card slot. The boards also provide two USB ports, a serial port and VGA output. Thus, convenient debugging of the nodes is even possible in case of network failure.

An on-board Ethernet interface with POE capability allows to run the node with a single cable plugged in and with a 100 Mbit Ethernet connection. Two miniPCI sockets are used with two Atheros AR5008 WLAN Cards. Multiple Input Multiple Output (MIMO) technology can be achieved by adding three antennas to each card. The authentication server and all MGs are connected over a 8-Port Gigabit Ethernet switch.

D. System Image and Software

Voyage Linux[1] is running as an operating system on all nodes, except for the meshctrl server which runs vanilla Debian 6.0. Voyage is a modified Debian Linux with optimizations for wireless drivers, the CF card and other hardware related issues. CF cards have very limited numbers of read and write cycles. Hence, the read and write actions must be reduced as much as possible. Voyage implements a temporary file system which writes each change to the file system into the RAM instead of the CF card. Only defined paths and files are written to the RAM, all other files are mounted read-only. This ensures a higher lifetime of the CF cards and thus a higher lifetime for the devices. All writable paths need to be specified in */etc/init.d/voyage-sync*.

The mesh routers are currently running a 3.2.9 Linux kernel. Important enabled kernel modules are B.A.T.M.A.N. [19] and the i2C module which allows to read values from the on-board temperature sensors. Communication with the kernel module is possible through the tool `batctl`. It allows to retrieve routing information, e.g., detected gateways, neighbors, translation and originator table, and visualization data. All interfaces that should use B.A.T.M.A.N. for routing are bridged to a *bat*-device. Batctl can also specify whether a device collects or sends visualization data.

Hostapd and wpa_supplicant are used for setting up the access points and the wireless connections to the mesh network. In a regular configuration of a node one of the WLAN cards is always connecting to another MR with wpa_supplicant and the other is offering an entry access point for other MRs by running a hostapd daemon. Hostapd can offer multiple virtual interfaces on one physical device. This allows to use one WLAN card not only as an entry point for other MRs, but also for MCs at the same time.

The two Atheros WLAN cards in each node are used with the *ath9k* [26] driver. For higher data rates and less interference with other wireless access points the testbed uses the 5GHz

[1] http://linux.voyage.hk/

band. Legacy entry points for 802.11b/g clients are possible too, but the mesh network traffic is sent using 802.11a [27].

E. Management

The process of running, maintaining, and configuring the ITsec testbed initially required a lot of manual effort. Additionally, monitoring the testbed was not possible at the time, rendering efficient maintenance even more difficult. In an evolving process, we developed a tailored tool called *pwrmesh*, which uses pwrcall[2] [28], a lightweight, secure bidirectional remote procedure call framework as a basis.

For management purposes we required to be able to change WiFi settings, switch regular nodes to gateway modus, reset/reboot nodes, configure networks settings (IP, iptables, ...), and add/remove users. Querying information about the nodes is important to be able to debug and restructure the testbed. Therefore, we are able to obtain the node's network state, e.g., IP, MAC, connected STAs, and its connectivity. In order to get information about our security mechanisms, we can query the IPsec SA lifetime and load, register SSID spoofing, manually trigger handover, and also analyze failed authentications. Lastly, we generated the network map using the `batmand` visualization and merge the information with our local static network topology map. Respective node maintenance we are able to flash new system images, run checks on the CF card and the node's memory, and reconfigure the PXE-Boot parameters.

We integrated all the above features using an agent-like setup. Each node runs the *pwrnode* which implements specific functions, i.e., obtaining the lifetime of IPsec SAs. All nodes are centrally managed by the *pwrserver* which is hosted on the meshctrl host. We created a Django[3] based web interface for controlling the pwrserver. Besides automatically rendering the information obtained from each pwrnode, we can also manually trigger commands on the nodes, e.g., rebooting from another PXE-Boot image, or setting up different wireless connections. Deploying patches for the most important software, i.e., hostapd, wpa_supplicant, and pwrnode, can also conveniently be done using our web interface.

Our secure management, maintenance, and monitoring is implemented as general as possible such that it can easily be applied to testbed setups different from ours. It can also be extended in a straightforward manner by implementing functionality on the nodes and the respective wrapper on the pwrserver, i.e., its web interface.

III. Experiments

This section shortly presents a selection of research and according experiments that have been enabled by our testbed.

A. ITsec Testbed Performance

In order to obtain the most important performance metrics of the ITsec testbed, we carried out measurements for packet loss, RTT, and throughput. All the measurements have been done using *iperf* and have been repeated a significant amount of times to obtain stable values. We have used an additional 5 MBit/s UDP noise stream on the same path, however, it only slightly influences the average packet loss. In terms of RTT, we compared a close to optimal Line of Sight (LOS) setup with the routers being regularly distributed throughout our institute. Obviously, RTTs increase as the overall distance increases. For the purpose of determining the throughput we ran a 20 seconds test using TCP. However, the optimal LOS setup produces a significantly larger amount of throughput.

B. Security Architecture

The security architecture of the testbed has been realized according to our prior research [29]. EAPs' Extended Master Session Key (EMSK), which is available at the AAA server, as well as at the authenticating nodes, is used as root in a hierarchy of keys. From the EMSK an Internet Protocol Security (IPsec) security association (containing an encryption key Traffic Encryption Key (TEK) and an integrity key Traffic Integrity Key (TIK)) is derived. If later on the node that joined the network acts as NAS, these keys are used to protect the authentication traffic between it and the AAA server with IPsec. The two remaining keys, the Peer Authentication Key (PAK) and the Key Derive Key (KDK), are used for authentication and key derivation during bootstrapping of the security associations required.

The Framework for establishing Security Associations for Sequentially Deployed WMN (FSASD) also allows to bootstrap security associations between any two authenticated nodes by using the 3-Party Handshake Protocol for Sequential Deployment (3PHSD) which interfaces with FSASD. The goal of 3PHSD is to allow any two already authenticated nodes A and B participating in the WMN to establish a security association with each other based on a key resulting from this protocol. In particular, 3PHSD can be used to set up an IPsec security association between MC and MG or to set up a link layer security association for CCMP between a moving MC and its new NAS during handover.

As for EAP-TTLS we were able to significantly improve its performance throughout our research. Running over UDP, the performance of EAP is almost proportional to the latency that can be measured on the path. After both wpa_supplicant and hostapd reached versions ≥ 1.0, the wireless performance of our testbed increased significantly. Also, the `setkey` command of the *ipsec-tools* package take some time.

C. Handover

In [30], we recently proposed three complementary secure, efficient, and practical proactive handover protocols, which are able to cope with the unique characteristics of WMNs such as the wireless infrastructure and untrusted intermediaries.

The goal of our protocols is to securely establish and transport a handover key, i.e., Pairwise Master Key (PMK) as known from IEEE 802.11i between the MC and the handover destination. Secure key transport between the involved parties was of paramount interest for us and is achieved by leveraging

[2]https://github.com/rep/pwrcall
[3]https://www.djangoproject.com/

prior work [29]. Once the MC decides to associate to another router, both can simply use the established PMK to carry out the 4-way handshake instead of running a lengthy and time consuming EAP authentication.

The first protocol, 3-Party Handshake Protocol for Handover (3PHSH), is a logical extension of 3PHSD as described in [29]. 3PHSH can be invoked at any time by the node requesting a handover, the handover target, and the AAA server. The other two protocols, Neighborhood Pre-Authentication (NPA) and EAP-TTLS Neighborhood Pre-Authentication (ENPA) allow an MC to prepare multiple possible handover destinations at once. ENPA leverages the EAP authentication of a device by including a list of potential handover targets. Upon successful authentication, the AAA generates individual handover keys and delivers them to the candidates requested by the MC. The last message of the EAP authentication from AAA to the MC contains the necessary parameters for the MC to be able to generate the handover keys. In terms of performance, embedding a list of target routers in the EAP messages using Diameter AVPs (Attribute Value Pairs) only slightly adds onto the overall duration of the EAP authentication.

All the protocols have been implemented as patches to wpa_supplicant and hostapd. For testing purposes we introduced a new command to manually trigger a handover to the *wpa_cli*, the command line interface of wpa_supplicant.

IV. Conclusion

In this paper we presented the ITsec Testbed as an experimentation platform for WMN research. We detailed the initial construction and our specific requirements towards a testbed for security research. The security architecture developed in [29] represents the cornerstone of the research that followed. In Section III we have shown research that has been sparked by the simple fact of a testbed being available. For instance, handover protocols for WMNs have, have to the best of our knowledge not been implemented and evaluated using a WMN testbed, yet. Our approach has shown that using off-the-shelf components facilitates building a testbed which enables researchers and students to obtain real world practical results which can complement often used simulations.

References

[1] A. Zimmermann, M. Günes, M. Wenig, U. Meis, and J. Ritzerfeld, "How to Study Wireless Mesh Networks: A hybrid Testbed Approach." in *AINA*. IEEE Computer Society, 2007.

[2] "KAUMesh - A Multi-Radio/Multi-Channel Wireless Mesh Testbed." [Online]. Available: {http://www.kau.se/en/kaumesh}

[3] H. Lundgren, K. Ramachandran, E. Belding-Royer, K. Almeroth, M. Benny, A. Hewatt, A. Touma, and A. Jardosh, "Experiences from the Design, Deployment, and Usage of the UCSB MeshNet Testbed," University of California, Santa Barbara, May 2012.

[4] "DES-Testbed at Freie Universität Berlin." [Online]. Available: {http://www.des-testbed.net/content/architecture}

[5] "freifunk.net: Free Networks, free WLAN." [Online]. Available: {http://wiki.freifunk.net/Kategorie:English}

[6] N. Ben Salem and J.-P. Hubaux, "Securing wireless mesh networks," *Wireless Communications, IEEE*, vol. 13, no. 2, pp. 50–55, 2006.

[7] O. Cheikhrouhou, M. Laurent-Maknavicius, and H. Chaouchi, "Security architecture in a multi-hop mesh networks," in *SAR'06*, 2006.

[8] Y. Zhang and Y. Fang, "ARSA: An Attack-Resilient Security Architecture for Multihop Wireless Mesh Networks." *IEEE Journal on Selected Areas in Communications*, 2006.

[9] M. S. Islam, Y. Y. Yoon, M. A. Hamid, and C. S. Hong, "A Secure Hybrid Wireless Mesh Protocol for 802.11s Mesh Network." in *ICCSA (1)*, ser. Lecture Notes in Computer Science. Springer, 2008.

[10] F. Martignon, S. Paris, and A. Capone, "MobiSEC: a novel security architecture for wireless mesh networks." in *Q2SWinet*, A. Y. Zomaya and M. Cesana, Eds. ACM, 2008.

[11] K. Ren and W. Lou, "A Sophisticated Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks." in *ICDCS*. IEEE Computer Society, 2008.

[12] M. Hamid, M. Islam, and C. seon Hong, "Developing Security Solutions for Wireless Mesh Enterprise Networks," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, 2008.

[13] R. Kandikattu and L. Jacob, "A Secure IPv6-based Urban Wireless Mesh Network (SUMNv6)." *Computer Communications*, 2008.

[14] L. Buttyan and L. Dora, "An Authentication Scheme for QoS-aware Multi-operator maintained Wireless Mesh Networks," in *IEEE WoWMoM*, 2009.

[15] K. Ren, S. Yu, W. Lou, and Y. Zhang, "PEACE: A Novel Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks," *Parallel and Distributed Systems, IEEE Transactions on*, 2010.

[16] B. He and D. Agrawal, "An Identity-based Authentication and Key Establishment Scheme for Multi-operator maintained Wireless Mesh Networks," in *IEEE MASS*, 2010.

[17] Z. Wang, M. Ma, W. Liu, and X. Wei, "A Unified Security Framework for Multi-domain Wireless Mesh Networks," in *ACM ICICS*, 2011.

[18] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "SAT: A Security Architecture Achieving Anonymity and Traceability in Wireless Mesh Networks," *Dependable and Secure Computing, IEEE Transactions on*, 2011.

[19] "B.A.T.M.A.N.: Better Approach to Mobile Ad-hoc Networking." [Online]. Available: {http://www.open-mesh.org/}

[20] K. Sollins, "The TFTP Protocol (Revision 2)," RFC 1350 (Standard), Internet Engineering Task Force, Jul. 1992, updated by RFCs 1782, 1783, 1784, 1785, 2347, 2348, 2349.

[21] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), Internet Engineering Task Force, Mar. 1997, updated by RFCs 3396, 4361, 5494. [Online]. Available: http://www.ietf.org/rfc/rfc2131.txt

[22] "hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator." [Online]. Available: http://hostap.epitest.fi/hostapd/

[23] "Linux WPA/WPA2/IEEE 802.1X Supplicant." [Online]. Available: http://hostap.epitest.fi/wpa_supplicant/

[24] "freeRADIUS: The world's most popular RADIUS Server." [Online]. Available: http://freeradius.org/

[25] P. Funk and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)," RFC 5281, Internet Engineering Task Force, Aug. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5281.txt

[26] "Atheros Linux wireless drivers: ath9k," Accessed August 24, 2012. Archieved at http://www.webcitation.org/6Ata833w4. [Online]. Available: {http://linuxwireless.org/en/users/Drivers/ath9k}

[27] I. C. Society, "Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band," *IEEE Std 802.11a*, 1999.

[28] M. Schloesser, "A new Framework for Secure Distributed Function Calls," Master's thesis, RWTH Aachen University, 2011.

[29] A. Egners, H. Fabelje, and U. Meyer, "FSASD: A framework for establishing security associations for sequentially deployed WMN." in *WOWMOM*. IEEE Computer Society, 2012.

[30] A. Egners, P. Herrmann, T. Jarmuzek, and U. Meyer, "Secure and Efficient Handover Protocols for WMNs," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE International Symposium on a*, june 2013.