Meta-Exploitation of IPv6-based WSNs

Riccardo Tomasi, Luca Bruno, Claudio Pastrone, Maurizio Spirito
Pervasive Radio Technologies Lab
Istituto Superiore Mario Boella
Torino, Italy
Email: tomasi@ismb.it, lucab@unstable.it, pastrone@ismb.it, spirito@ismb.it

***Abstract*—The growing use of Wireless Sensor Networks (WSNs) to accomplish tasks with strict requirements in terms of secrecy and reliability is increasingly attracting the interest of malicious users, thus incrementing the demand for more secure and dependable WSN implementations. Such necessity is directly reflected in the growing need for tools and techniques to ease the design, assessment and validation of security solutions suited for WSN deployments. Initially, WSNs were mainly considered as stand-alone systems but the emerging Internet of Things (IoT) vision is fostering a deeper integration of WSNs as a part of the Internet, hence making holistic approaches and re-use of existing solutions more effective also concerning security. In order to cope with this evolving scenario, this paper proposes a tool to support penetration testing of actual WSN deployments based on the 6LoWPAN family of protocols, considered one of the main building blocks of the IoT paradigm. The feasibility of WSN penetration testing as an approach to tackle IoT security is demonstrated through a practical proof-of-concept developed by extending the well-known Metasploit Framework targeting actual WSN deployments based on 6LoWPAN.**

I. Introduction

Wireless Sensor Networks (WSNs) are self configuring ad-hoc networks composed of wirelessly connected devices, usually characterized by constrained computational and memory resources, low consumption, unmanned and unattended operations. In the past, WSNs have been often considered as "stand-alone" systems. In many actual deployments, WSNs are in fact composed of nodes relying on custom protocols and algorithms, forming a self-consistent and autonomous network. The boundaries of such WSNs are clearly defined and represented by one or more "sink" nodes integrated with a dedicated server-side application or gateway.

However, the rise of the Internet of Things (IoT) [1] is rapidly transforming the WSN scenario. Because of the IoT vision, the use of WSNs is rapidly moving towards an Internet-like paradigm, in which devices can inter-operate seamlessly with each other and with other heterogeneous systems through the adoption of open protocols [1]. Due to its scalability, flexibility and compatibility with existing solutions [2], the IPv6 [3] is expected to play a key role in the paradigm shift from stand-alone WSNs to Internet-connected, large-scale Low-power Wireless Personal Area Networks (LoWPANs). Nevertheless, IPv6 is not suitable to be employed "as-is" in WSNs and thus needs adaptation layers such as IPv6 over LoW Power Wireless Area Networks (6LoWPAN) [4], [5], [6]. LoWPAN technologies are being increasingly adopted as components of complex mainstream applications (e.g. home automation, industrial automation, energy monitoring) even within security-sensitive tasks. Consequently, the spread of systems adopting LoWPAN technologies is likely to cause a proportional increment of interest by malicious users.

The research community has just recently started to tackle attacks to WSN deployments [7] and, more in general, LoWPAN security [8] both from the theoretical and practical side. Thus, a lot of research work is still needed in the area of assessment, testing and improvement of LoWPAN security. Practical methods such as penetration testing [9] could represent a viable approach to fill, at least partially, this gap. Penetration testing is an offensive security approach in which evaluators impersonate an attacker provided with realistic resources and knowledge about the target system. During the emulated attack, evaluators generally attempt to modify the normal behavior of the target system to gain resources (e.g. access, information) through illegitimate means, trying to disrupt or circumvent security features.

Practical tests using a similar approach have been already performed targeting existing WSN implementations [10]. However, such attacks were performed through custom-made, platform-specific tools, and are not thus fully suitable to target a LoWPAN characterized by standard protocols.

A currently available example of existing tool-set for penetration testing targeting LoWPAN technologies is instead represented by KillerBee [11]. KillerBee is python-based and supports security attacks to ZigBee networks, leveraging on a custom firmware installed on-board an AVR RZ Raven USB Stick transceiver device. KillerBee features have been proven through a series of demonstration including ZigBee traffic sniffing, network stumbling (dumping summarized information about ZigBee networks active within the local transmission range), replay attacks, and even physical exploits e.g. looking for network shared key in the memory dump from a "stolen" device. However, KillerBee features are specific to ZigBee and thus not suitable to address 6LoWPAN deployments.

A wide number of solutions are instead available to perform penetration testing against systems and networks employing

Acknowledgments: The authors gratefully acknowledge partial funding of this work by the regional project "Piattaforma Tecnologica Innovativa per l'Internet of Things", and the collaboration between ISMB and Politecnico di Torino within the FP7 Network of Excellence Newcom++ (contract no. 216715). **Disclaimer:** All techniques described in this paper are for educational and research purposes only and must not be used against actual deployments. The authors can not be considered responsible for any damage, direct or indirect, caused by the described solution.

IPv6 [12] including: network sniffers e.g. Wireshark [13]; network scanners e.g. NMap [14]; security scanners e.g. Nessus [15], OpenVAS [16]; and other general-purpose tools e.g. Socat [17]. Many of these tools are used passively to gather intelligence about the system under test, while others are used more actively to perform practical attacks.

Among this last category, a widely used solution is represented by the Metasploit Framework (MF) delivered by the Metasploit Project. The MF is a Ruby-based collection of tools and libraries allowing the modeling, development and execution of attacks against the system under test [18].

Attacks targeting IPv6 implementations are already available in the MF [19] which has also already been used to develop attacks targeting proprietary short-range wireless solutions e.g. wireless presenters [20]. However the MF is not yet able to target networks based on 6LoWPAN or other protocols employing IEEE 802.15.4 such as ZigBee [21].

In conclusion, despite relevant penetration testing examples and tools targeting either WSNs or IPv6-based systems have already been demonstrated, a comprehensive penetration testing framework targeting IPv6-enabled WSNs has not been defined, yet.

In order to help filling such gap, this paper proposes an extension of the MF enabling penetration testing of IPv6-enabled WSNs i.e. 6LoWPAN-based networks. The idea of extending an existing framework instead of developing a new tool from scratch is supported by different arguments. Firstly, the MF is already provided with attack automation capabilities improved over the years by the development community. Secondly, the MF is already capable of performing attacks targeting IPv6-based networks: the proposed extension would allow to handle LoWPAN technologies too. Finally, this choice would pave the way towards future integration with other security tools, which are already used in synergy with the MF.

The remaining of the paper is organized as follows. Sec. II provides background information about 6LoWPAN and its typical usage scenarios. Sec. III describes the proposed architecture i.e. how the MF is extended to cope with 6LoWPAN technologies. Sec. IV provides a proof-of-concept by describing an attack example performed using the proposed solution. Finally, Sec. V draws conclusions.

II. 6LoWPAN Background

6LoWPAN [4], [5], [22], [6] is a protocol under definition within the Internet Engineering Task Force (IETF) [23] and backed by the Internet Protocol for Smart Objects (IPSO) Alliance [24]. The main goal of 6LoWPAN is to provide IPv6 connectivity to Low-power, Lossy Networks (LLNs) based on IEEE 802.15.4 [25].

LLNs can be defined as a generalization of WSNs whose focus is not strictly on data collection: as WSNs, LLNs are typically composed of a possibly large number of low-power nodes communicating wirelessly in ad-hoc, multi-hop fashion by means of short-sized radio packets. According to the 6LoWPAN paradigm, each LLN node becomes an IPv6 enabled host.

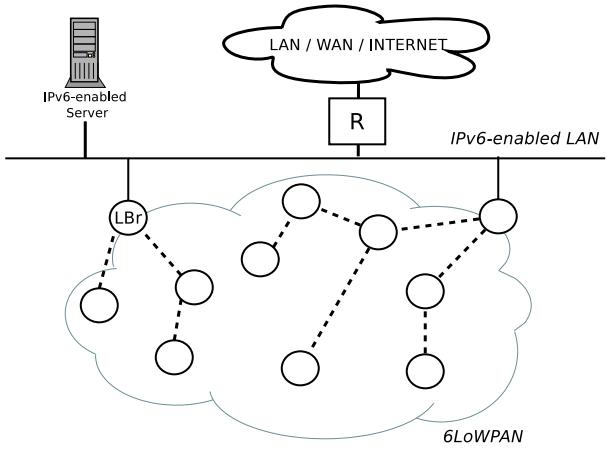

Fig. 1. A 6LoWPAN reference architecture

As already mentioned in Sec. I, IPv6 can not be used "as-is" in such constrained networks, due to limitations and specific features of IEEE 802.15.4 (e.g. the minimum IPv6 MTU (Maximum Transmission Unit exceeds the maximum IEEE 802.15.4 packet size); as a result, different adaptations must be employed. Such adaptations include specific approaches for sub-packet fragmentation [5], header compression [26], [5], neighbor discovery [27], routing [28] and transport/application [29].

Fig. 1 depicts a common network architecture which can be employed by a 6LoWPAN-based WSN. As for WSNs, 6LoWPAN nodes are able to self-configure, forming autonomously a wireless multi-hop network. Such network is bridged to a wired IPv6-enabled LAN by means of one or more LoWPAN Border routers (LBRs) converting 6LoWPAN frames into full IPv6 packets. LBRs are used provide the LoWPAN with connectivity from and to a local IPv6-enabled LAN, which can in turn provide access to some larger-scale networks (e.g. the Internet) through normal IPv6 routers. LoWPAN nodes can be either deployed in fixed positions or move and can have different characteristics in terms of processing power, sensing capabilities and energy sources. According to the IoT philosophy, a certain degree of flexibility is left to the managers of the LoWPAN deployment: data locally collected by sensors can be both forwarded remotely and processed locally, depending on the application. Local processing can be performed in cooperative fashion leveraging on short-range data exchange between neighboring nodes and can be used e.g. to enrich raw data collected by the sensors. In such a scenario the LoWPAN is expected to be subject to a number of security requirements [8] e.g. Confidentiality, Integrity, Authentication, Freshness, Availability, Robustness, etc. Enforcing the aforementioned security requirements in the described context is expected to be a complex and cumbersome task: the network under discussion is in fact "hybrid", being both a WSAN and a IPv6-enabled network. Due to such hybrid nature, the LoWPAN is subject to security issues encompassing both the worlds of IPv6 and WSN security.

IPv6 security is, above all, Internet security: all security considerations are based on the Internet scenario, its protocols and its usage/service model. Concerning the IPv6 protocol itself, a number of possible security threats are already known [12]; they are due either to the IPv6 procedures e.g. stateless auto-configuration (SLAAC), Duplicate Address Detection (DAD), etc. or to extensions such as e.g. Mobile IPv6 or IPv4/IPv6 transition support.

Concerning WSN security, a number of more complex and diverse issues must be taken into account [30]. Firstly, as descibed in Fig. 1, the LoWPAN segment of the architecture communicates through a shared medium: this entails that a possible attacker using a radio device in promiscuous mode (P) is able to overhear and inject packet in the network, if communication is not secured. Secondly, the way WSNs operate introduces vulnerabilities and complexity in the security model: many WSN features (e.g. multi-hop communication) are in fact based on the assumption that all nodes are cooperative and trustworthy [30]; however, in real-world applications normally no mechanism is provided to enforce an adequate level of trust. Thirdly, WSNs may be left unattended in extremely large scale environments, enlarging the physical attack surface (e.g. risk of physical theft, tampering). Finally, WSNs are characterized by several limitations in terms of processing power, available energy, radio link reliability, delays: such constraints make WSN nodes "weak" almost by design. Such limitations not only constitute a security weakness, but might as well obstacle the set-up and operation of countermeasures [8] as supported by several examples such as (1) the difficulty to run asymmetric cryptography routines on-board typical WSN nodes due to the relatively high energy consumption and the relatively low storage space and processing power; (2) the difficulty to run a practical authentication method or key exchange scheme over unreliable multi-hop paths characterized by high latency and high packet loss. Consequently, the appropriateness of any WSN security countermeasure is directly connected to its sustainability i.e. the ability of being security-effective without hindering normal WSN operations.

Concerning practical attacks, due to the relative scarcity of 6LoWPAN implementation in use in real-world deployments, a lot of research work is still to be done.

However, a significantly large number of attacks [8] at different ISO/OSI layers have already been identified as feasible: physical layer attacks, including different "flavors" of radio jamming, eavesdropping/sniffing, injection-based attacks; data link layer attacks including MAC collision, exhaustion and unfairness, or attacks based on ACK frame integrity; network layer attacks including routing information spoofing, selective forwarding, sinkhole attacks, Sybil attacks, wormhole attack, attacks towards neighbor discovery mechanisms.

III. Proposed MF Extension

Following the analysis in Sec. II, the 6LoWPAN segment could easily represent the weakest security link of an IoT architecture, lowering the overall level of security. In order to tackle such issue, security aspects in 6LoWPAN segment should be carefully investigated and considered at design time. Moreover, technical information security testing and assessments should be performed on the final system to verify whether specific security objectives are met. With respect to security testing and assessment, this paper proposes a solution which extends the MF with the possibility to perform penetration tests targeting such segment.

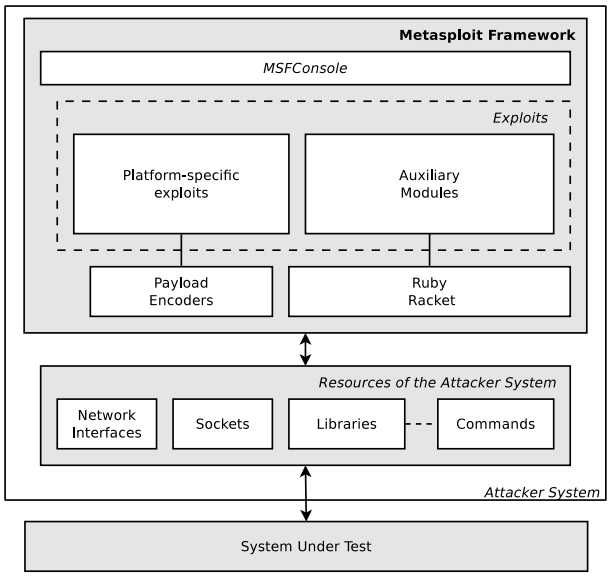

Fig. 2. 6LoWPAN exploits within the MF

The MF components are ideally structured as shown in Fig. 2. The most significant part of the MF is represented by a large set of modules called "exploits". Exploits are Ruby scripts which emulate the behavior of a human attacker, pursuing the goal of automatically assessing (or breaking) the security of one or more target *systems under test*.

MF exploits can be classified in two categories: *platform-specific exploits* and *auxiliary modules*.

Platform-specific exploits leverage on hardware or software vulnerabilities including e.g. attacks based on buffer overflow. The final goal is to cause the system under test to execute the "payload" which is a piece of software forged to allow execution of arbitrary code, architecture-dependent and generated through Payload Encoders.

Auxiliary modules instead target vulnerabilities or features of protocols and do not need any platform-specific encoding [31]. Auxiliary modules are normally used for a number of purposes including e.g. network scanning, packet filtering, computations, etc.

Both platform-specific exploits and auxiliary modules directly leverage on resource of the attacker system, eg. sockets and network tools. In addition, auxiliary modules also strongly rely on lower level primitives for *protocol dissection* and *protocol encoding*, the former usually associated with packet sniffing and the latter with packet injection.

Within the MF such functions are supported by *Ruby Racket* [32], a Ruby framework which allows to encode and

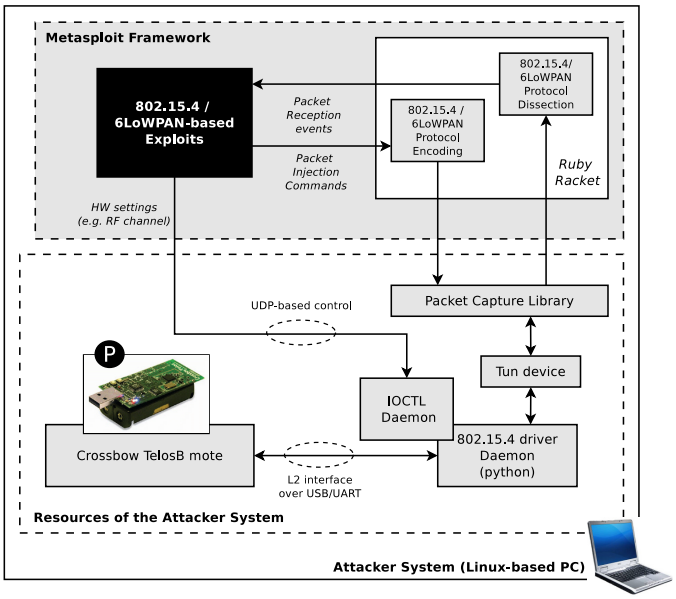

Fig. 3. Architecture of a simple test-bed demonstrating the proposed solution

decode packets using a predefined tree of protocols. Within Ruby Racket, protocols are modeled through a collection of classes divided in communication layers roughly matching the ISO communication stack. Once a packet is processed (or generated) by Ruby Racket it is represented by a *packet* object containing multiple *layer* children objects. Any protocol field is reflected by a simple attribute of the layer object and it can thus be read and modified as a simple variable, ignoring field format issues such as endianness, sign or size of the field. MF exploits, driven both in automated or semi-automated fashion, can be configured at runtime and launched by means of a control console i.e. MSFconsole, which also provides output about the status of ongoing attacks.

The proposed MF extension is composed of a set of modules aiming at: (1) exposing system-wide interfaces for network interaction, (2) supporting encoding/decoding of the IEEE 802.15.4 and 6LoWPAN protocols and (3) providing auxiliary modules to actually drive 6LoWPAN-specific attacks.

The architecture of the resulting extended MF is described in Fig. 3.

The attacker is expected to run an MF instance e.g. on normal laptop running Linux, equipped with an IEEE 802.15.4-compliant transceiver (*P*), sniffing any packet transmitted over-the-air within its reception range. Sniffed packets are encapsulated and sent over a layer-2 (L2) interface by a IEEE 802.15.4 network interface driver running on the attacker system. From there, packets are received by the MF (but also other tools e.g. Wireshark [13]) through a Packet Capture Library (libpcap) and processed by Ruby Racket. Within the proposed solution, pyhsical radio interface has been implemented using the well-known Crossbow Telos rev. B mote hardware [33]; since a network driver for the Linux kernel supporting IEEE 802.15.4 and the aforementioned hardware is not yet available, such component has been implemented with a Python-based daemon which exposes to the system a virtual L2 network device (TUN/TAP), accessible by libpcap.

For easier development of exploits based on IEEE 802.15.4 and 6LoWPAN, any of the fields of these protocols should be easily accessible and modifiable at runtime. In order to make this possible, Ruby Racket dictionaries for IEEE 802.15.4 and 6LoWPAN packets protocol have been implemented. A Data Link-level dictionary accommodate different cases specified by IEEE 802.15.4 standard, including data, acknowledgement, beacon and command frame, as well as optional and variable fields. On top of it, a set of 6LoWPAN modules process data packets according to RFC 4944 specifications and header compression schemes, currently at draft stage but already implemented by some OS, eg. TinyOS. The additional modules, glued together with already implemented IPv6 and ICMPv6 classes, offer a full stack to properly dissect/forge entire packets, from IEEE 802.15.4 to any upper layers.

As described in Fig. 3, when a IEEE 802.15.4 packet from *P* is made available inside the libpcap, it can be decoded by the Ruby Racket decoders, which make available a new packet object to any running exploits. In case the IEEE 802.15.4 packet is a 6LoWPAN packet, the 6LoWPAN layer of the packet will also be populated by the decoder.

Running exploits can filter the incoming decoded packets based on their fields and react accordingly e.g. by forging and injecting a new packet. For example, an exploit can be instructed to filter any packet matching some conditions (i.e. 6LoWPAN/IPv6 packets whose next header is ICMPv6 and whose ICMPv6 type is 133) and selectively answer with a forged packet. Forged packets, in turn, can be populated according the format of any protocol for whom an encoder is available (e.g. an ICMPv6 Destination Unreachable packet). Fields can be seamlessly copied from incoming packets to outgoing packets being forged (e.g. setting as destination the source IPv6 address of the incoming packet).

It is also possible to alter some field in a packet which has just been sniffed and re-inject immediatly: this can be useful for man-in-the-middle attacks. Other types of attacks can be developed leveraging on the APIs exposed by the MF: for example, a simple infinite loop sending continuosly raw 802.15.4 packets could be a starting point to develop a Jamming-based attack.

As a front-end part of the proposed solution, a number of auxiliary attack scripts have been developed to monitor and poison a 6LoWPAN network, e.g. by creating fake node identities and corrupting default routes.

IV. A SIMPLE SINKHOLE-BASED ATTACK

In order to assess its functionalities,the proposed solution has been employed to drive an attack against a small-scale 6LoWPAN testbed composed of a small set of nodes running the TinyOS BLIP stack [34] and a simple monitoring application. The implemented attack pattern follows the well-known Sinhole attack pattern [35]. Such attack represents a simple

and understandable proof-of-concept, useful to provide a better insight about the functionalities of the proposed solution.

Despite it may not seem reasonable to target an unsecured network, it is worth observing that many current 6LoWPAN deployments do not support yet any advanced security feature. In fact, many of these features are still under definition and, in some deployments, they are not even applicable due to hardware limitations of WSN nodes.

The attacker controls a single promiscuous node *P* through an instance of the proposed solution running e.g on a normal laptop. The transmission range of *P* overlaps with the transmission ranges of one or more LoWPAN nodes. In a realisitc attack the first goal is to gather intelligence about active devices and their operations. The proposed solution supports these operations by eavesdropping over-the-air communication as a network detector, sniffer and logger. In order to avoid being detected by Intrusion Detection Systems (IDS) possibly operating within the network, the attacker starts the attack with a passive phase.

In first place, the attacker hops through the 16 available IEEE 802.15.4 channels, listening for any traffic. Once some 6LoWPAN activity is detected, the attacker selects the relevant operating channel and starts sniffing and storing all IEEE 802.15.4 traffic.

By filtering the sensed traffic, the attacker is able to collect knowledge about the LoWPAN including active networks and their topology, the list of active nodes and their role (e.g. whether they are acting as routers), various network and node identifiers (WPAN identifiers, IPv6 subnetwork prefix, IEEE 802.15.4 short address,...), etc.

Sensed information may also include possible unencrypted sensitive data or portions of cryptographic materials to feed future cracking attacks.

Once the attacker has gathered sufficient knowledge, by leveraging on packet injection capabilities, he gains the possibility to indirectly impact on a wider scale by manipulating protocols and algorithms that build and maintain the network topology. If even a single legitimate link of the LoWPAN is not secured, in fact, due to the in-band signaling used in ad-hoc networks, it becomes relatively easy to hijack unprotected network resources or disrupt communications. Within this scenario, the Sinkhole attack is realized by intercepting unsecured on-link Router Solicitations and Advertisements, and then by injecting forged packets to poison routing and SLAAC features.

In order for this attack to scale up to more than one target node, the attacker (P) must be within the transmission range of at least one router node. Once an ICMPv6 Router Solicitation is detected, P triggers a routine that generates a 6LoWPAN-encapsulated Router Advertisement with a fake source router address. After the forged advertisement is sent over-the-air, it will indirectly alter the routing tables of all neighboring nodes, which will immediately adopt the new node as default router as depicted in Fig. 4.

Such alteration will indirectly hijack traffic flows from entire sections of the network, or possibly from the whole network

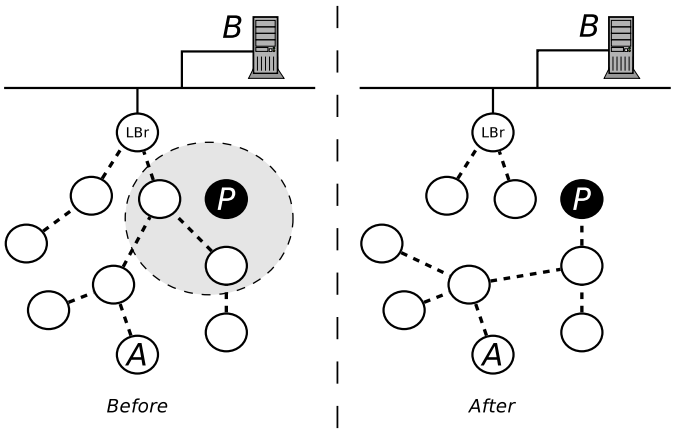

Fig. 4. Attack results

if the legitimate border router is misconfigured or attacked by a Denial of Service (DoS) attack from another P node. If configuration from authoritative routers is not authenticated nor protected by other means, the results of the procedure will be an IPv6 subnet hijacking.

As depicted in Fig. 4, for instance, as a result of such attack, the Attacker can gain an "in the middle" position between the fixed IPv6-enabled LAN and an entire portion of the LoWPAN.

Moreover, if P has enough resources also to forward collected traffic towards/from a legitimate router, it can assume the role of a transparent "evil" proxy, enabling further man-in-the-middle attacks based on selective filtering and forging of packets.

V. Conclusions

Penetration testing tools and methods represent a practical solution to assess and test security weaknesses of actual LoWPAN deployments, which may easily represent the weakest security link of an IoT architecture. The result of such testing should provide useful elements to LoWPAN designers and thereby make systems and protocol more robust and trustworthy.

In order to demonstrate the feasibility of such penetration testing approaches, the original contribution of this paper is an extension of the well-known Metasploit Framework to support attacks targeting IPv6 enabled WSNs. The resulting 6LoWPAN enabled penetration testing tool has been validated through a proof-of-concept attack targeting an actual small-scale WSN testbed.

The presented attack is very simple and targets an unsecured network: although it is not fully significant from the security assessment point of view, it has been used to provide more understanding about how the proposed solution operates.

The proposed solution can target networks in any scenario where the penetration tester can take control of at least one promiscuous WSN node within the radio range of a WSN. In many scenarios (e.g. fire detection, area monitoring,...) this happens very easily, due to the large scale operations.

Additionally, the decoding capabilities of the proposed solution are hindered by encrypted communication. However, due to the difficulty in setting up and operating practically any WSN security countermeasure, a large amount of the currently deployed WSNs are not secured, enlarging the number of possible attacks.

Except for these limitations, it would rather simple for a tester to inject false events (e.g. fire alarms) or to disrupt WSN operation so to prevent important data to be delivered (e.g. a trespassing warning). Nevertheless, the proposed approach remains valid also against networks where countermeasures (e.g. encryption) are in place, although with minor impact (e.g. detection of network activity, collection of material for future cracking attacks,...).

Future works may employ the developed solution to support more complex attack scenarios known in literature listed in Sec. II, so to assess more advanced security features described in current and upcoming 6LoWPAN IETF drafts. Future activities should also evaluate robustness and reliability of the proposed solution.

REFERENCES

[1] O. Ovidiu Vermesan, M. Harrison, H. Vogt, K. Kalaboukas, M. Tomasella, K. Wouters, S. Gusmeroli, S. Haller *et al.*, "Internet of things," CERP-IoT - Cluster of European Research Projects on the Internet of Things, Strategic Research Roadmap, 2009.

[2] A. Dunkels and J. Vasseur, "IP for Smart Objects," Internet Protocol for Smart Objects Alliance, IPSO White Paper 1, 2008.

[3] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460 (Draft Standard), Internet Engineering Task Force Std. 2460, Dec. 1998, updated by RFCs 5095, 5722, 5871. [Online]. Available: http://www.ietf.org/rfc/rfc2460.txt

[4] N. Kushalnagar, G. Montenegro, and C. Schumacher, *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, RFC 4919 (Informational), Internet Engineering Task Force Std. 4919, Aug. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4919.txt

[5] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC 4944 (Proposed Standard), Internet Engineering Task Force Std. 4944, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4944.txt

[6] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet (Wiley Series on Communications Networking & Distributed Systems)*, Wiley, Ed. Wiley, 2010.

[7] P. Radmand, A. Talevski, S. Petersen, and S. Carlsen, "Taxonomy of wireless sensor network cyber security attacks in the oil and gas industries," vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 949–957.

[8] S. Park, K. Kim, W. Haddad, S. Chakrabarti, and J. Laganier, *IPv6 over Low Power WPAN Security Analysis*, Internet-Draft (work in progress), draft-daniel-6lowpan-security-analysis-04, Internet Engineering Task Force Std., 2010. [Online]. Available: http://tools.ietf.org/html/draft-daniel-6lowpan-security-analysis-04

[9] J. Wack, M. Tracy, and M. Souppaya, "Guideline on Network Security Testing," NIST Special Publication, Tech. Rep. 800-42, 2003.

[10] F. Stajano, D. Cvrcek, and M. Lewis, "Steel, cast iron and concrete: Security engineering for real world wireless sensor networks," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, S. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds. Springer Berlin / Heidelberg, 2008, vol. 5037, pp. 460–478.

[11] J. Wright. (2010, Accessed Dec.) Killerbee, framework and tools for exploiting ZigBee and IEEE 802.15.4 networks. http://code.google.com/p/killerbee/.

[12] E. Martin, "Ipv6 Security," Cisco Systems, Inc., Technical talk, May 2003, available at http://www.6net.org/events/workshop-2003/marin.pdf.

[13] The Wireshark Foundation. (2010, Accessed Dec.) Wireshark network protocol analyzer. http://www.wireshark.org.

[14] G. Lyon *et al.* (2010, Accessed Dec.) Nmap. http://www.nmap.org.

[15] Tenable Network Security. (2010, Accessed Dec.) Nessus. http://www.nessus.org.

[16] The OpenVAS project. (2010, Accessed Dec.) Openvas. http://www.openvas.org.

[17] The Socat project. (2010, Accessed Dec.) Socat multipurpose relay. http://www.dest-unreach.org/socat/.

[18] The Metasploit Project. (2010, Accessed Dec.) The Metasploit Framework. http://www.metasploit.com.

[19] The Metasploit project. (2010, Accessed Dec.) Exploiting IPv6. http://www.metasploit.com/data/confs/ - sector2008/exploiting_ipv6.pdf.

[20] N. Teusink. (2010, Accessed Dec.) Hacking wireless presenters with an Arduino and Metasploit. http://blog.teusink.net/2010/07/- hacking-wireless-presenters-with.html.

[21] ZigBee Standards Organization, *ZigBee Specification*, ZigBee Alliance Std., 2008.

[22] J. Hui, D. Culler, and S. Chakrabarti, "6LoWPAN: incorporating IEEE 802.15.4 into the IP architecture," Internet Protocol for Smart Objects Alliance, IPSO White Paper 3, 2009.

[23] The Internet Engineering Task Force (IETF). (2010, Accessed Dec.) The IETF web site. http://www.ietf.org.

[24] Internet Protocol for Smart Objects Alliance. (2010, Accessed Nov.) The IPSO Alliance web site. http://www.ipso-alliance.org.

[25] IEEE, *Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., 2006.

[26] J. W. Hui and P. Thubert, *Compression Format for IPv6 Datagrams in 6LoWPAN Networks*, Internet-Draft (work in progress), draft-ietf-6lowpan-hc-13, Internet Engineering Task Force Std., 2010. [Online]. Available: http://tools.ietf.org/html/draft-ietf-6lowpan-hc-13

[27] Z. Shelby, S. Chakrabarti, and E. Nordmark, *Neighbor Discovery Optimization for Low-power and Lossy Networks*, Internet-Draft (work in progress), draft-ietf-6lowpan-nd-14, Internet Engineering Task Force Std., 2010. [Online]. Available: http://tools.ietf.org/html/draft-ietf-6lowpan-nd-14

[28] T. Winter and P. Thubert, *RPL: IPv6 Routing Protocol for Low power and Lossy Networks*, Internet-Draft (work in progress), draft-ietf-roll-rpl-12, Internet Engineering Task Force Std., 2010. [Online]. Available: http://tools.ietf.org/html/draft-ietf-roll-rpl-12

[29] Z. Shelby, B. Frank, and D. Sturek, *Constrained Application Protocol (CoAP)*, Internet-Draft (work in progress), draft-ietf-core-coap-02, Internet Engineering Task Force Std., 2010. [Online]. Available: http://tools.ietf.org/html/draft-ietf-core-coap-02

[30] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey,âĂİ in book chapter of security," in *in Distributed, Grid, and Pervasive Computing, Yang Xiao (Eds*. CRC Press, 2007, pp. 0–849.

[31] D. Kennedy *et al.* (2010, Accessed Dec.) Metasplot unleashed tutorial. http://www.offensive-security.com/- metasploit-unleashed/.

[32] J. Hart. (2010, Accessed Dec.) Ruby racket. http://spoofed.org/files/-racket/doc/.

[33] Memsic, *Telos rev.B datasheet*, Accessed Dec. 2010, available at http://www.memsic.com/products/- wireless-sensor-networks/wireless-modules.html.

[34] S. Dawson-Haggerty *et al.* (2010, Accessed Dec.) The BLIP stack for TinyOS 2.x. http://smote.cs.berkeley.edu:8000/- tracenv/wiki/blip.

[35] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Launching a sinkhole attack in wireless sensor networks; the intruder side," in *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 526–531. [Online]. Available: http://portal.acm.org/citation.cfm?id=1475703.1476542