Protocol Reverse Engineering Methods for Undocumented Ethernet and Wireless Protocols; *Survey*

Baraka D Sija, Young-Hoon Goo, Kyu-Seok Shim and Myung-Sup Kim+
Department of Computer and Information Science, Korea Univ.
{sijabarakajia25, gyh0808, kusuk007, tmskim}@korea.ac.kr

Abstract

A protocol is a set of rules that govern communications between two or more machines in a network whereas Protocol Reverse Engineering (PRE) is the process of uncovering protocol specifications of both known and unknow protocols. Due to network and Internet security threats, nowadays Protocol Reverse Engineering is even becoming more important and highly demanded. However, Protocol Reverse Engineering yet faces major limits and challenges yet unsolved, such as complexity, tedious and error-prone, time consuming and manual (requiring a lot of human intervention). In this paper, we survey two papers of two different approaches in Protocol Reverse Engineering, an approach to Reverse Engineer Ethernet(LAN) protocols and an approach to Reverse Engineer IEEE 802.15.4 wireless protocols. While Protocol Reverse Engineering on Ethernet(LAN) protocols have been highly studied, few studies and few works have been proposed for Reverse Engineering wireless protocols. Therefore, this paper discuses *WASp,* an approach that analyzes and reconstructs unknown wireless customized protocols over IEEE 802.15.4 in details. Another approach discussed is *AutoFormat,* an approach that extracts protocol fields and reveal the inherently "non-flat", hierarchical structures of protocol messages with a summarized introduction to PRE.

Keywords: Network Protocols, Protocol Reverse Engineering, IEEE 802.15.4 wireless protocols, Ethernet Protocols, Survey

Ⅰ. Introduction

Protocol Reverse Engineering is the process in which protocol parameters, formats and semantics are inferred in the absence of formal specifications. Since protocol specifications of unknown protocols are mainly kept secret by developers or owners the only way to uncover the specifications is through reverse engineering. Many reasons may apply to why the inventors are likely to hide the specifications of the protocol they develop, but all pertaining to pursuing individual or organization profits while some protocols in the Internet are developed solely for surveillance. Such kind of protocols are hard to uncover their specifications. Moreover, protocols are constantly evolving due to features and functionalities changes that lead to even harder and more complicated protocol reverse engineering process. Although automatic Protocol Reverse Engineering methods are being proposed and developed, the process has been mostly manually done, which is error-prone and time consuming. Accuracy maximization, automation and time shortening for the PRE process are the priorities for the modern security communication environment.

In an Internet of Things (IoT) environment, many devices communicate with one another using various wireless communication protocols. Because most applications operate in a personal area such as a home, office, or hospital, wireless personal area network (WPAN) technologies, such as ZigBee, Bluetooth, and Z-Wave, are essential for implementing IoT systems [1]. Therefore, the specifications of protocols involved during such wireless IoT communications need to be uncovered as IEEE.802.15.4 has been used for lower layer protocols and number of wireless protocols are still unknown and undocumented.

Ⅱ. Motivation

Tools like Wireshark, TCPdump and Microsoft Network Monitor have the capability of understanding and classifying many protocols, however there are so many variations and subprotocols that exist in real network environment making it difficult for such tools to uncover them all completely. Forgetting about Ethernet protocols, what is even harder for such tools is uncovering of wireless protocols. This paper, discusses two PRE approaches, *AutoFormat* and *WASp*, however *WASp* which focuses on reverse engineering IEEE 802.15.4 wireless protocols for spoofing is the main interest and motivator of this work.

Ⅲ. WASp

WPAN automatic spoofer (WASp) is a tool based on wireless customized protocols over IEEE 802.15.4, which is capable of understanding and reconstructing customized protocols to byte-level accuracy, and to generate packets that can be used for verification of analysis results or spoofing attacks. IEEE 802.15.4 is a standard that defines lower layer protocols for WPAN communications that use frequency bands of 868– 868.6, 902– 928, or 2,400– 2,483.5 MHz. It mainly focuses on low-speed and low-power

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2015R1D1A3A01018057) and Science Technology and Humanities Convergence Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning(No.NRF-2016M3C1B6929228).

communication between wireless devices. Although Wi-Fi offers high-speed solutions, IEEE 802.15.4 requires much less power and focuses on home-range devices [1]. It supports a communication range of 10 m, and a transfer rate of 250 kbps. Technological simplicity and flexibility make IEEE 802.15.4. adaptable to various wireless communication protocols. For instance, well-known protocols, such as ZigBee, 6LoWPAN, WirelessHART, have been developed based on the IEEE 802.15.4 standard.

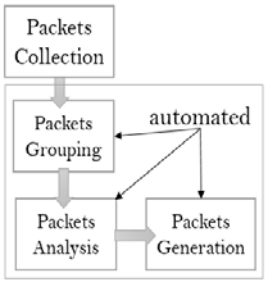

Fig. 1. WASp: Summarized architecture overview

As depicted in a summarized figure 1, WASp consists of four phases: *packet collection, packet grouping, protocol analysis, and packet generation*, whereby packets collection is conducted manually with the rest modules being fully automated. Manually packet collection refers to wireless channel sniffing process that must be conducted in carefully controlled conditions. Packets collection affects the effectiveness of analysis because critical factors such as the number of collected packets, the number of transceivers, and expected operations are bounded in this phase. Initially, WASp groups packets according to crucial information from packet headers, for each packet group it analyzes MAC layer data reuse such as address field, byte-level entropy, the range of each byte column, and the existence of a cyclic redundancy check (CRC) [1]. In the second step, WASp combines results of all tests and generates scored analysis reports for every packet group using a scoring algorithm. Conclusively, WASp analyzes reconstructs unknown wireless customized protocols over IEEE 802.15.4 for spoofing.

IV. AutoFormat

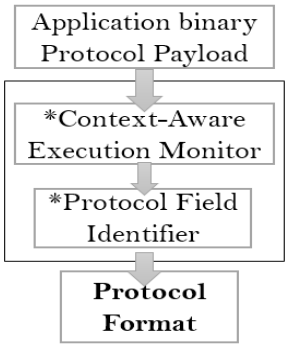

Fig. 2. AutoFormat: Summarized Architecture overview

AutoFormat is a system that aims at extracting protocol fields with high accuracy and revealing the inherently "non-flat", hierarchical structures of protocol messages. AutoFormat is based on the key insight that different protocol fields in the same message are typically handled in different execution contexts (e.g., the runtime call stack). As such, by monitoring the program execution, AutoFormat collects the execution context information for every message byte (annotated with its offset in the entire message) and clusters them to derive the protocol format. AutoFormat carried an evaluation with more than 30 protocol messages from seven protocols, including two text-based protocols (HTTP and SIP), three binary-based protocols (DHCP, RIP, and OSPF), one hybrid protocol (CIFS/SMB), as well as one unknown protocol used by a real-world malware.

AutoFormat is interested in how field-specific execution context information can be collected and analyzed to extract protocol format. Figure 2 shows a summarized architectural overview of AutoFormat, which has two starred main components: *a context-aware execution monitor* and *a protocol field identifier*. When AutoFormat receives a binary that implements the protocol to be analyzed, it functions as follows, first; on receiving an incoming protocol message, it marks the received data and keeps track of their propagation at the byte granularity, second; once a message byte is read, the execution monitor logs that byte, its offset in the entire message, and the run-time execution context at that moment, which includes the call stack and the location of the instruction being executed. Third, with the collected context information, the protocol field identifier which runs offline is invoked to identify protocol fields and extract the structural layout of the message.

IV. Conclusion and Future Work

This paper, discusses two tools, WASp (a tool for RE an IEEE 802.15.4 wireless protocols for spoofing) and AutoFormt (a tool for RE protocol formats from application binaries. Although more research and new automated PRE techniques are needed, much has been done for Reverse Engineering Ethernet protocols compared to wireless protocols. From this work, the need to research and RE wireless protocols has been discovered as our future work.

References

[1] Kibum Choi, Yunmok Son, Juhwan Noh, Hocheol Shin, Jaeyeong Choi, Yongdae Kim, "Dissecting Customized Protocols: Automatic Analysis for Customized Protocols based on IEEE 802.15.4", Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, July 18-20, 2016, Darmstadt, Germany

[2] Z. Lin, X. Jiang, D. Xu, and X. Zhang, "Automatic Protocol Format Reverse Engineering through Context-Aware Monitored Execution", In 15th Symposium on Network and Distributed System Security (NDSS), February 2008.