802.11p Literature Review and Attacks Content Draft

Aniket Chaudhari and Alvin Cai and Wouter de Groot and Erik Schneider
Technische Universiteit Eindhoven, Eindhoven, The Netherlands

***Abstract*—In this content-draft, we outline their project to understand and attack 802.11p in the context of Security and Privacy in Mobile Systems.**

I. Introduction

802.11p is part of the 802.11 wireless communication standards family designed for vehicle to vehicle communication. It is based on 802.11a with the major design differences to accommodate communication links that might exist for only a short amount of time.

802.11p itself is a catch-all term to refer to the set of applications for which it is used. As such, our research efforts will not only be limited to the MAC and PHY areas themselves. Rather, we will also look at the protocols on top of it.

In this paper we will provide a literature review on 802.11p and some existing attacks. We will evaluate through practical experiments the feasibility of some of these attacks. Finally, we will attempt to discover new attacks through the implementation of a fuzzing framework.

II. Background

The background section will contain the literature survey. Most likely it will consist of a section on the basics of 802.11p and the protocols that run on top of it, and another section on the existing attacks. Our preliminary intent is to focus on basic denial of service (DOS) and packets in packets [**?**].

III. Existing Attacks

We will attempt to recreate the attacks by interfacing with the Atheros ath5k drivers (for packets in packets attack) or modifying the ath5k drivers (for DOS attack). We will then evaluate if the 802.11p reference devices (available at Twente) or Atheros drivers are vulnerable.

Finally, if the drivers are vulnerable, we will attempt to propose a fixe and evaluate the efficiacy of this fix.

IV. Fuzzing Framework

Undiscovered implementation bugs in the 802.11p drivers might be present as it is a relatively new protocol (not yet mainstream), device drivers are typically written in C code where it is easy to make mistakes and device drivers are also potentially less audited than mainline kernel codes [**?**]. In this context, fuzzing is the most ideal approach to discover vulnerabilities as it has a good earnings to price ratio to discover vulnerabilities as compared to the more tedious code review.

In this project, we will decide on a fuzzing architecture which includes the hardware setup, software tools to use (i.e. leveraging existing tools such as Scapy, develop custom tools or leveraging tools used in Chapter **??**) and finally, the 802.11p protocol fields to fuzz.

V. Results

This section will contain a list of exploits discovered as a direct result of our fuzzing. For the purposes of this content-draft we use the standard IMRAD layout, but if in reality we manage to find one or more weaknesses we anticipate this section will be restructured to introduce the flaw, detail the method of its exploitation and finally to explain the potential consequences of exploitation.

VI. Analysis

This section may be merged with Results if we come up with an actual attack. If our hard work does not result in practical attacks, we will likely use this area to analyze our approach.

VII. Discussion

In the closing section it is time to reflect on our work. How did our work contribute to the state of the art? What are the consequences for 802.11p? Which are, in our opinion and the opinions of the reviewed literature, promising avenues of research? What are the limitations in our research?

References

[1] Goodspeed, Travis, et al. "Packets in Packets: Orson Welles' In-Band Signaling Attacks for Modern Radios." WOOT. 2011.

[2] Butti, Laurent, and Julien Tinns. "Discovering and exploiting 802.11 wireless driver vulnerabilities." Journal in Computer Virology 4.1 (2008): 25-37.