Active Automata Learning of an IPsec IKEv1 Server using AALpy

Benjamin Wunderling[1][0000−1111−2222−3333]

TU Graz IST, Graz 8010, Austria
benjamin.wunderling@student.tugraz.at

Abstract. Virtual Private Network (VPN) protocols are widely used to create a secure mode of communication between several parties over an insecure channel. A common use-case for VPNs is to secure access to company networks. Therefore errors in VPN software are often severe. IPsec is a VPN protocol that uses the Internet Key Exchange protocol (IKE). IKE has two versions, IKEv1 and the newer IKEv2. While several papers have investigated IPsec-IKEv2 in the context of Automata learning, no such work has been performed for IPsec-IKEv1. This short paper describes the IPsec-IKEv1 protocol and show the steps taken to learn the state-machine of an IPsec server. We present a learned model and discuss its potential applications for model-based fuzzing and fingerprinting of IPsec implementations.

Keywords: IPsec · Automata Learning · AALpy.

1 Introduction

VPNs are used to allow secure communication over an insecure channel. The importance of VPN software has increased dramatically since the beginning of the COVID-19 pandemic due to the influx of people working from home [1]. This makes finding vulnerabilities in VPN software more critical than ever. IPsec is a VPN protocol and most commonly uses the IKE protocol to share authenticated keying material between involved parties. Therefore, IKE and IPsec are sometimes used interchangeably. We will stick to the official nomenclature of using IPsec for the full protocol and IKE for the key exchange only. IKE has two versions, IKEv1 IKEv2, with IKEv2 being the newer and recommended version [3]. However, despite IKEv2 supposedly replacing its predecessor, IKEv1 is still in widespread use today. This is reflected by the company AVM to this day only offering IKEv1 support for their popular Fritzbox routers [4].

State models of protocol implementations are useful tools in testing. They can for example be used to detect software versions [6], or generate test cases automatically [7]. One method of generating such models is to use Active Automata Learning. A notable example of an Active Automata Learning algorithm is the L* algorithm by Angluin [2]. In L*, a teacher queries the System under Learning (SUL) and through its responses can construct an automaton describing the behavior of the SUL. This automaton can then be compared with the

SUL, adapting it if they show different behaviors. Several papers have investigated IPsec-IKEv2 using Automata Learning, however so far none have looked at IKEv1.

We show the process of learning a state model from an example IPsec-IKEv1 server. We use the Active Automata Learning framework AALpy [5] for L* Automata Learning, with a custom python interface between AALpy and the IPsec server.

In this short paper we first go over preliminary information on VPNs and Automata Learning in chapter 2. In 3 we discuss other related work. In chapter 4 we briefly introduce AALpy and our learning setup. Then we will present our custom interface between AALpy an the IPsec server, discussing design choices and implementation difficulties. Finally we present the learned model and discuss its potential applications and further work in chapters 5 and 6.

2 Preliminaries

2.1 Automata Learning

2.2 IPsec

3 Related Work

4 Learning IPsec

4.1 Setup

4.2 AALpy

4.3 Our Interface

5 Evaluation

6 Conclusion

6.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.
Displayed equations are centered and set on a separate line.

$$x + y = z \tag{1}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. **??**).

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [**?**], an LNCS chapter [1], a book [**?**], proceedings without editors [**?**], and a homepage [**?**]. Multiple citations are grouped [**?**,**?**,**?**], [**?**,**?**,**?**,**?**].

Acknowledgements Please place your acknowledgments at the end of the paper, preceded by an unnumbered run-in heading (i.e. 3rd-level heading).

References

1. Abhijith, M., Senthilvadivu, K.: Impact of vpn technology on it industry during covid-19 pandemic. In: IJEAST (2020)
2. Angluin, D.: Learning regular sets from queries and counterexamples. Information and computation **75**(2), 87–106 (1987)
3. Barker, E., Dang, Q., Frankel, S., Scarfone, K., Wouters, P.: Guide to ipsec vpns (2020-06-30 00:06:00 2020). https://doi.org/https://doi.org/10.6028/NIST.SP.800-77r1
4. GmbH, A.C.V.: Connecting the fritz!box with a company's vpn. https://en.avm.de/service/vpn/tips-tricks/connecting-the-fritzbox-with-a-companys-vpn/, accessed: 2022-09-09

5. Muškardin, E., Aichernig, B.K., Pill, I., Pferscher, A., Tappler, M.: Aalpy: an active automata learning library. Innovations in Systems and Software Engineering pp. 1–10 (2022)
6. Pferscher, A., Aichernig, B.K.: Fingerprinting bluetooth low energy devices via active automata learning. In: International Symposium on Formal Methods. pp. 524–542. Springer (2021)
7. Pferscher, A., Aichernig, B.K.: Stateful black-box fuzzing of bluetooth devices using automata learning. In: NASA Formal Methods Symposium. pp. 373–392. Springer (2022)