Laboratory Report
Rohit Raj

The EAP-TLSv1.3 connection behavior

1 Introduction

EAP-TLS is widely considered as one of strongest EAP methods for wireless enterprise authentication. It uses Transport Layer Security protocol and unlike other implementations, it requires x509 client-side certificates. The latest version of Transport Layer Security (TLS) 1.3 was released in August 2018. This release has brought some exciting new features along with improved latency and security during connections. These new features affects the way current EAP-TLS authentications are handled. Our goal of this experiment was to analyze the behavior of different EAP-TLS 1.3 implementations(namely *HostApd* and *FreeRadius*).

1.1 Experimental Setup

The setup comprised of the following three EAP components:

- *Supplicant* : The client attempting EAP-TLS connections was a Ubuntu 18.03 machine. For making the EAP-TLS connections, *wpa-supplicant* was used.
- *Authenticator* : *D-LINK DWL-6600AP* router acted as an authenticator and this access point was configured for EAP-TLS communication along with the address of Authentication server on one of its SSID.
- *Authentication Server* : Our choice of Authentication server were popular flavors of RADIUS server such as *FreeRadius* and *HostApd*. The server itself resided on Cloud and received the RADIUS packets from our Authenticator on port 1802.

2 HostAp TLS 1.3 behavior

3 Burning Questions!

1. Will the PSK be common for all the devices?

2. Will the key renegotiation be a part of TLS1.3 architecture, or are we supposed to do something else?

3. Starting point of implementation. *wpa_supplicant* or *hostap*? At what point will we deviate from standard EAP-TLS (purely in implementational sense)?