A Brief Architectural Overview to *xsupplicant*
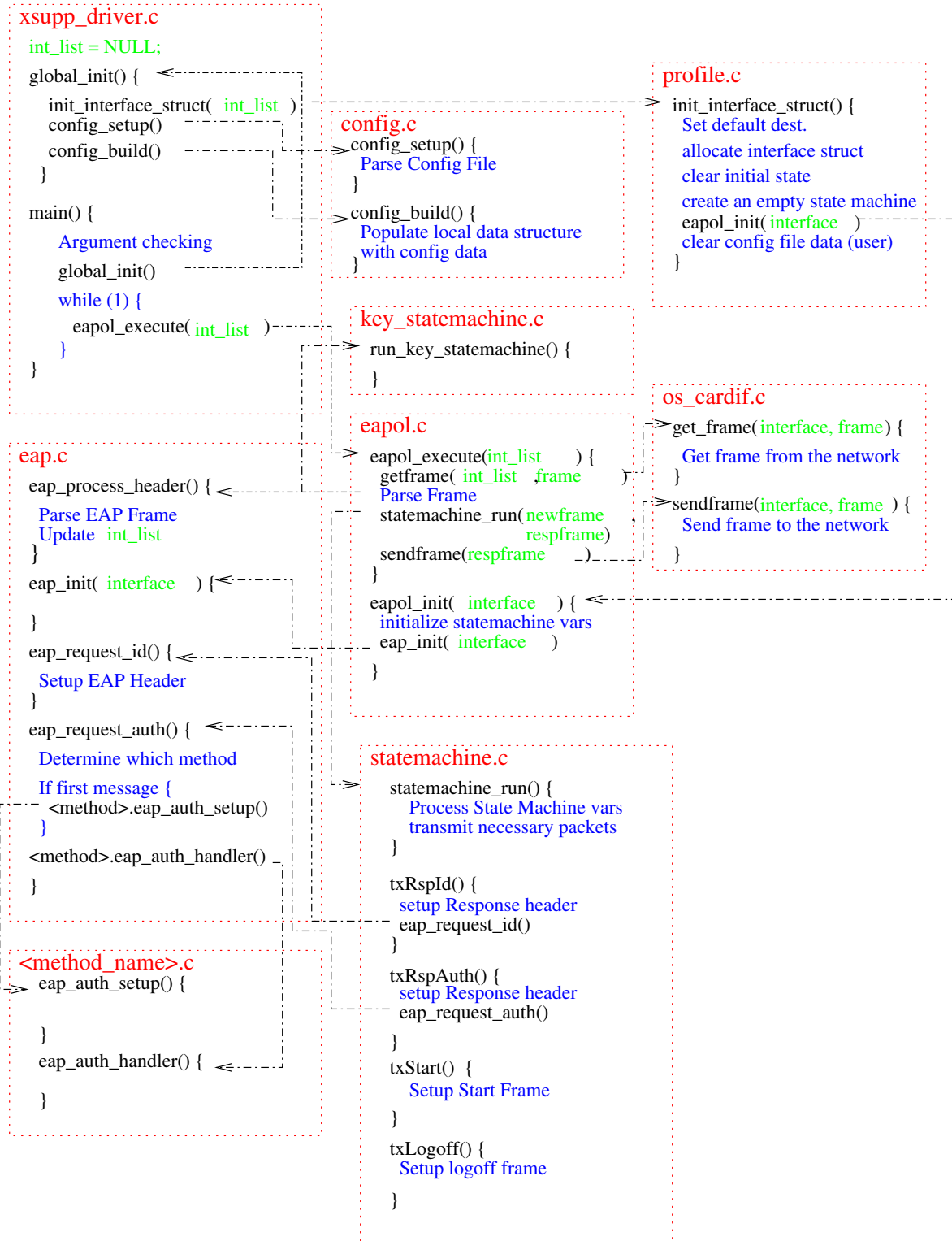
The Xsupp Guys

November 24, 2003

1 Introduction

This document serves as an overview of the code structure of *xsupplicant*. It's intended purpose is to agree on a model to code to such that future bug fixes, features, and other updates can be discussed in a more structured manner. This document, like *xsupplicant*, is a work in progress.

2 Current Code Flow

The diagram below attempts to capture the current flow of information through the *xsupplicant* code. Note that this assumes normal operation. Error conditions are outside the scope of this section, unless explicitly stated. The following notation is used below:

- File names are in red.
- Function names are in black.
- Pseudocode is in blue.
- Data is in green.

xsupp_driver.c

int_list = NULL;

global_init() {
  init_interface_struct(int_list)
  config_setup()
  config_build()
}

main() {
  Argument checking
  global_init()
  while (1) {
    eapol_execute(int_list)
  }
}

config.c

config_setup() {
  Parse Config File
}

config_build() {
  Populate local data structure
  with config data
}

profile.c

init_interface_struct() {
  Set default dest.
  allocate interface struct
  clear initial state
  create an empty state machine
  eapol_init(interface)
  clear config file data (user)
}

key_statemachine.c

run_key_statemachine() {

}

eapol.c

eapol_execute(int_list) {
  getframe(int_list ,frame)
  Parse Frame
  statemachine_run(newframe
               respframe)
  sendframe(respframe)
}

eapol_init(interface) {
  initialize statemachine vars
  eap_init(interface)
}

os_cardif.c

get_frame(interface, frame) {
  Get frame from the network
}

sendframe(interface, frame) {
  Send frame to the network
}

eap.c

eap_process_header() {
  Parse EAP Frame
  Update int_list
}

eap_init(interface) {

}

eap_request_id() {
  Setup EAP Header
}

eap_request_auth() {
  Determine which method
  If first message {
    <method>.eap_auth_setup()
  }
  <method>.eap_auth_handler()
}

statemachine.c

statemachine_run() {
  Process State Machine vars
  transmit necessary packets
}

txRspId() {
  setup Response header
  eap_request_id()
}

txRspAuth() {
  setup Response header
  eap_request_auth()
}

txStart() {
  Setup Start Frame
}

txLogoff() {
  Setup logoff frame
}

<method_name>.c

eap_auth_setup() {

}

eap_auth_handler() {

}

3 Data Structures

This section provides and overview to the various data structures used in xsupplicant. This is an overview, but is not guaranteed to be in sync with the latest code. Do not rely on this document for coding, just for

an overview of how things are structured.