RaspAP / **raspap-webgui** Public

<> Code ⊙ Issues 24 Pull requests 4 Discussions Actions Projects ···

b02660d5ff ▾ ···

raspap-webgui / includes / **hostapd.php** / <> Jump to ▾

billz Rename ambiguous variable for clarity History

8 contributors

Executable File | 376 lines (341 sloc) | 15.6 KB ···

```php
<?php

require_once 'status_messages.php';
require_once 'app/lib/system.php';
require_once 'includes/wifi_functions.php';
require_once 'includes/config.php';

getWifiInterface();

/**
 *
 *
 */
function DisplayHostAPDConfig()
{
    $status = new StatusMessages();
    $system = new System();
    $arrConfig = array();
    $arr80211Standard = [
        'a' => '802.11a - 5 GHz',
        'b' => '802.11b - 2.4 GHz',
        'g' => '802.11g - 2.4 GHz',
        'n' => '802.11n - 2.4 GHz',
        'ac' => '802.11.ac - 5 GHz'
    ];
    $arrSecurity = array(1 => 'WPA', 2 => 'WPA2', 3 => 'WPA+WPA2', 'none' => _("None"));
    $arrEncType = array('TKIP' => 'TKIP', 'CCMP' => 'CCMP', 'TKIP CCMP' => 'TKIP+CCMP');
    $managedModeEnabled = false;
    exec("ip -o link show | awk -F': ' '{print $2}'", $interfaces);

    if (!RASPI_MONITOR_ENABLED) {
        if (isset($_POST['SaveHostAPDSettings'])) {
            SaveHostAPDConfig($arrSecurity, $arrEncType, $arr80211Standard, $interfaces, $status);
        }
    }

    $arrHostapdConf = parse_ini_file('/etc/raspap/hostapd.ini');

    if (!RASPI_MONITOR_ENABLED) {
         if (isset($_POST['StartHotspot']) || isset($_POST['RestartHotspot'])) {
            $status->addMessage('Attempting to start hotspot', 'info');
            if ($arrHostapdConf['BridgedEnable'] == 1) {
                exec('sudo /etc/raspap/hostapd/servicestart.sh --interface br0 --seconds 3', $return);
            } elseif ($arrHostapdConf['WifiAPEnable'] == 1) {
                exec('sudo /etc/raspap/hostapd/servicestart.sh --interface uap0 --seconds 3', $return);
            } else {
                exec('sudo /etc/raspap/hostapd/servicestart.sh --seconds 3', $return);
            }
            foreach ($return as $line) {
                $status->addMessage($line, 'info');
            }
        } elseif (isset($_POST['StopHotspot'])) {
            $status->addMessage('Attempting to stop hotspot', 'info');
            exec('sudo /bin/systemctl stop hostapd.service', $return);
            foreach ($return as $line) {
                $status->addMessage($line, 'info');
            }
        }
    }

    exec('cat '. RASPI_HOSTAPD_CONFIG, $hostapdconfig);
    exec('iwgetid '. $_POST['interface']. ' -r', $wifiNetworkID);
    if (!empty($wifiNetworkID[0])) {
        $managedModeEnabled = true;
    }
    $hostapdstatus = $system->hostapdStatus();
    $serviceStatus = $hostapdstatus[0] == 0 ? "down" : "up";

    foreach ($hostapdconfig as $hostapdconfigline) {
        if (strlen($hostapdconfigline) === 0) {
            continue;
        }

        if ($hostapdconfigline[0] != "#") {
            $arrLine = explode("=", $hostapdconfigline);
            $arrConfig[$arrLine[0]]=$arrLine[1];
        }
    };

    echo renderTemplate(
        "hostapd", compact(
            "status",
            "serviceStatus",
            "hostapdstatus",
            "managedModeEnabled",
            "interfaces",
            "arrConfig",
            "arr80211Standard",
            "selectedHwMode",
            "arrSecurity",
            "arrEncType",
            "arrHostapdConf"
        )
    );
}

function SaveHostAPDConfig($wpa_array, $enc_types, $modes, $interfaces, $status)
{
    // It should not be possible to send bad data for these fields so clearly
    // someone is up to something if they fail. Fail silently.
    if (!(array_key_exists($_POST['wpa'], $wpa_array)
        && array_key_exists($_POST['wpa_pairwise'], $enc_types)
        && array_key_exists($_POST['hw_mode'], $modes))
    ) {
        error_log("Attempting to set hostapd config with wpa='".$_POST['wpa']."', wpa_pairwise='".$_POST['wpa_pairwise']."' and hw_mode='".$_POST['hw_mode']."'");  // FIXME: log i
        return false;
    }

    if (!filter_var($_POST['channel'], FILTER_VALIDATE_INT)) {
        error_log("Attempting to set channel to invalid number.");
        return false;
    }

    if (intval($_POST['channel']) < 1 || intval($_POST['channel']) > 48) {
        error_log("Attempting to set channel to '".$_POST['channel']."'");
        return false;
    }

    $good_input = true;

    // Check for Bridged AP mode checkbox
    $bridgedEnable = 0;
    if ($arrHostapdConf['BridgedEnable'] == 0) {
        if (isset($_POST['bridgedEnable'])) {
            $bridgedEnable = 1;
        }
    } else {
        if (isset($_POST['bridgedEnable'])) {
            $bridgedEnable = 1;
        }
    }

    // Check for WiFi client AP mode checkbox
    $wifiAPEnable = 0;
    if ($bridgedEnable == 0) {  // enable client mode actions when not bridged
        if ($arrHostapdConf['WifiAPEnable'] == 0) {
            if (isset($_POST['wifiAPEnable'])) {
                $wifiAPEnable = 1;
            }
        } else {
            if (isset($_POST['wifiAPEnable'])) {
                $wifiAPEnable = 1;
            }
        }
    }

    // Check for Logfile output checkbox
    $logEnable = 0;
    if ($arrHostapdConf['LogEnable'] == 0) {
        if (isset($_POST['logEnable'])) {
            $logEnable = 1;
            exec('sudo '.RASPI_CONFIG.'/hostapd/enablelog.sh');
        } else {
            exec('sudo '.RASPI_CONFIG.'/hostapd/disablelog.sh');
        }
    } else {
        if (isset($_POST['logEnable'])) {
            $logEnable = 1;
            exec('sudo '.RASPI_CONFIG.'/hostapd/enablelog.sh');
        } else {
            exec('sudo '.RASPI_CONFIG.'/hostapd/disablelog.sh');
        }
    }

    $cfg = [];
    $cfg['WifiInterface'] = $_POST['interface'];
    $cfg['LogEnable'] = $logEnable;
    // Save previous Client mode status when Bridged
    $cfg['WifiAPEnable'] = ($bridgedEnable == 1 ?
        $arrHostapdConf['WifiAPEnable'] : $wifiAPEnable);
    $cfg['BridgedEnable'] = $bridgedEnable;
    $cfg['WifiManaged'] = $_POST['interface'];
    write_php_ini($cfg, RASPI_CONFIG.'/hostapd.ini');
    $_SESSION['ap_interface'] = $_POST['interface'];

    // Verify input
    if (empty($_POST['ssid']) || strlen($_POST['ssid']) > 32) {
        // Not sure of all the restrictions of SSID
        $status->addMessage('SSID must be between 1 and 32 characters', 'danger');
        $good_input = false;
    }

    if ($_POST['wpa'] !== 'none'
        && (strlen($_POST['wpa_passphrase']) < 8 || strlen($_POST['wpa_passphrase']) > 63)
    ) {
        $status->addMessage('WPA passphrase must be between 8 and 63 characters', 'danger');
        $good_input = false;
    }

    if (isset($_POST['hiddenSSID'])) {
        if (!is_int((int)$_POST['hiddenSSID'])) {
            $status->addMessage('Parameter hiddenSSID not a number.', 'danger');
            $good_input = false;
        } elseif ((int)$_POST['hiddenSSID'] < 0 || (int)$_POST['hiddenSSID'] >= 3) {
            $status->addMessage('Parameter hiddenSSID contains invalid configuratie value.', 'danger');
            $good_input = false;
        } else {
            $ignore_broadcast_ssid = $_POST['hiddenSSID'];
        }
    } else {
        $ignore_broadcast_ssid = '0';
    }

    if (! in_array($_POST['interface'], $interfaces)) {
        // The user is probably up to something here but it may also be a
        // genuine error.
        $status->addMessage('Unknown interface '.htmlspecialchars($_POST['interface'], ENT_QUOTES), 'danger');
        $good_input = false;
    }

    if (strlen($_POST['country_code']) !== 0 && strlen($_POST['country_code']) != 2) {
        $status->addMessage('Country code must be blank or two characters', 'danger');
        $good_input = false;
    }

    $_POST['max_num_sta'] = (int) $_POST['max_num_sta'];
    $_POST['max_num_sta'] = $_POST['max_num_sta'] > 2007 ? 2007 : $_POST['max_num_sta'];
    $_POST['max_num_sta'] = $_POST['max_num_sta'] < 1 ? null : $_POST['max_num_sta'];

    if ($good_input) {
        // Fixed values
        $country_code = $_POST['country_code'];
        $config = 'driver=nl80211'.PHP_EOL;
        $config.= 'ctrl_interface='.RASPI_HOSTAPD_CTRL_INTERFACE.PHP_EOL;
        $config.= 'ctrl_interface_group=0'.PHP_EOL;
        $config.= 'auth_algs=1'.PHP_EOL;
        $config.= 'wpa_key_mgmt=WPA-PSK'.PHP_EOL;
        $config.= 'beacon_int=100'.PHP_EOL;
        $config.= 'ssid='.$_POST['ssid'].PHP_EOL;
        $config.= 'channel='.$_POST['channel'].PHP_EOL;
        if ($_POST['hw_mode'] === 'n') {
            $config.= 'hw_mode=g'.PHP_EOL;
            $config.= 'ieee80211n=1'.PHP_EOL;
            // Enable basic Quality of service
            $config.= 'wmm_enabled=1'.PHP_EOL;
        } elseif ($_POST['hw_mode'] === 'ac') {
            $config.= 'hw_mode=a'.PHP_EOL.PHP_EOL;
            $config.= '# N'.PHP_EOL;
            $config.= 'ieee80211n=1'.PHP_EOL;
            $config.= 'require_ht=1'.PHP_EOL;
            $config.= 'ht_capab=[MAX-AMSDU-3839][HT40+][SHORT-GI-20][SHORT-GI-40][DSSS_CCK-40]'.PHP_EOL.PHP_EOL;
            $config.= '# AC'.PHP_EOL;
            $config.= 'ieee80211ac=1'.PHP_EOL;
            $config.= 'require_vht=1'.PHP_EOL;
            $config.= 'ieee80211d=0'.PHP_EOL;
            $config.= 'ieee80211h=0'.PHP_EOL;
            $config.= 'vht_capab=[MAX-AMSDU-3839][SHORT-GI-80]'.PHP_EOL;
            $config.= 'vht_oper_chwidth=1'.PHP_EOL;
            $config.= 'vht_oper_centr_freq_seg0_idx=42'.PHP_EOL.PHP_EOL;
        } elseif ($_POST['hw_mode'] === 'w') {
            $config.= 'ieee80211w=2'.PHP_EOL;
            $config.= 'wpa_key_mgmt=WPA-EAP-SHA256'.PHP_EOL;
        } else {
            $config.= 'hw_mode='.$_POST['hw_mode'].PHP_EOL;
            $config.= 'ieee80211n=0'.PHP_EOL;
        }
        if ($_POST['wpa'] !== 'none') {
            $config.= 'wpa_passphrase='.$_POST['wpa_passphrase'].PHP_EOL;
        }
        if ($wifiAPEnable == 1) {
            $config.= 'interface=uap0'.PHP_EOL;
        } elseif ($bridgedEnable == 1) {
            $config.='interface='.$_POST['interface'].PHP_EOL;
            $config.= 'bridge=br0'.PHP_EOL;
        } else {
            $config.= 'interface='.$_POST['interface'].PHP_EOL;
        }
        $config.= 'wpa='.$_POST['wpa'].PHP_EOL;
        $config.= 'wpa_pairwise='.$_POST['wpa_pairwise'].PHP_EOL;
        $config.= 'country_code='.$_POST['country_code'].PHP_EOL;
        $config.= 'ignore_broadcast_ssid='.$ignore_broadcast_ssid.PHP_EOL;
        if (isset($_POST['max_num_sta'])) {
            $config.= 'max_num_sta='.$_POST['max_num_sta'].PHP_EOL;
        }

        file_put_contents("/tmp/hostapddata", $config);
        system("sudo cp /tmp/hostapddata " . RASPI_HOSTAPD_CONFIG, $return);

        // Fetch dhcp-range, lease time from system config
        $dhcpConfig = parse_ini_file(RASPI_DNSMASQ_CONFIG, false, INI_SCANNER_RAW);

        if ($wifiAPEnable == 1) {
            // Enable uap0 configuration in dnsmasq for Wifi client AP mode
            // Set dhcp-range from system config. If undefined, fallback to default
            $dhcp_range = ($dhcpConfig['dhcp-range'] =='10.3.141.50,10.3.141.255,255.255.255.0,12h' ||
                $dhcpConfig['dhcp-range'] =='') ? '192.168.50.50,192.168.50.150,12h' : $dhcpConfig['dhcp-range'];
            $config = 'interface=lo,uap0               # Enable uap0 interface for wireless client AP mode'.PHP_EOL;
            $config.= 'bind-dynamic                    # Hybrid between --bind-interfaces and default'.PHP_EOL;
            $config.= 'server=8.8.8.8                  # Forward DNS requests to Google DNS'.PHP_EOL;
            $config.= 'domain-needed                   # Don\'t forward short names'.PHP_EOL;
            $config.= 'bogus-priv                      # Never forward addresses in the non-routed address spaces'.PHP_EOL;
            $config.= 'dhcp-range='.$dhcp_range.PHP_EOL;
            if (!empty($dhcpConfig['dhcp-option'])) {
                $config.= 'dhcp-option='.$dhcpConfig['dhcp-option'].PHP_EOL;
            }
        } else {
            // Set dhcp-range from system config. If undefined, fallback to default
            $dhcp_range = ($dhcpConfig['dhcp-range'] =='192.168.50.50,192.168.50.150,12h' ||
                $dhcpConfig['dhcp-range'] =='') ? '10.3.141.50,10.3.141.255,255.255.255.0,12h' : $dhcpConfig['dhcp-range'];
            $config = 'domain-needed'.PHP_EOL;
            $config.= 'interface='.$_POST['interface'].PHP_EOL;
            $config.= 'dhcp-range='.$dhcp_range.PHP_EOL;
            if (!empty($dhcpConfig['dhcp-option'])) {
                $config.= 'dhcp-option='.$dhcpConfig['dhcp-option'].PHP_EOL;
            }
        }
        file_put_contents("/tmp/dnsmasqdata", $config);
        system('sudo cp /tmp/dnsmasqdata '.RASPI_DNSMASQ_CONFIG, $return);

        // Set dnsmasq values from ini, fallback to default if undefined
        $intConfig = parse_ini_file(RASPI_CONFIG_NETWORKING.'/'.$_POST['interface'].'.ini', false, INI_SCANNER_RAW);
        $domain_name_server = ($intConfig['domain_name_server'] =='') ? '1.1.1.1 8.8.8.8' : $intConfig['domain_name_server'];
        $routers = ($intConfig['routers'] == '') ? '10.3.141.1' : $intConfig['routers'];

        // write options to dhcpcd.conf
        $config = [ '# RaspAP '.$_POST['interface'].' configuration' ];
        $config[] = 'hostname';
        $config[] = 'clientid';
        $config[] = 'persistent';
        $config[] = 'option rapid_commit';
        $config[] = 'option domain_name_servers, domain_name, domain_search, host_name';
        $config[] = 'option classless_static_routes';
        $config[] = 'option ntp_servers';
        $config[] = 'require dhcp_server_identifier';
        $config[] = 'slaac private';
        $config[] = 'nohook lookup-hostname';

        if ($bridgedEnable == 1) {
            $config[] = 'denyinterfaces eth0 wlan0';
            $config[] = 'interface br0';
        } elseif ($wifiAPEnable == 1) {
            // Enable uap0 configuration in dhcpcd for Wifi client AP mode
            $intConfig = parse_ini_file(RASPI_CONFIG_NETWORKING.'/uap0.ini', false, INI_SCANNER_RAW);
            $ip_address = ($intConfig['ip_address'] == '') ? '192.168.50.1/24' : $intConfig['ip_address'];
            $config[] = 'interface uap0';
            $config[] = 'static ip_address='.$ip_address;
            $config[] = 'nohook wpa_supplicant';
        } else {
            // Default config
            $ip_address = "10.3.141.1/24";      // fallback IP
            // default IP of the AP xxx.xxx.xxx.1/24 of the selected dhcp range
            $def_ip = array();
            if (preg_match("/^([0-9]{1,3}\.){3}/",$dhcp_range,$def_ip) ) $ip_address = $def_ip[0]."1/24";
            // use static IP assigned to interface only, if consistent with the selected dhcp range
            if (preg_match("/^([0-9]{1,3}\.){3}/",$intConfig['ip_address'],$int_ip) && $def_ip[0] === $int_ip[0]) $ip_address = $intConfig['ip_address'];
            $config[] = 'interface '.$_POST['interface'];
            $config[] = 'static ip_address='.$ip_address;
            $config[] = 'static domain_name_server='.$domain_name_server;
            $config[] = PHP_EOL;

            // write the static IP back to the $_POST['interface'].ini file
            $intConfig['interface'] = $_POST['interface'];
            $intConfig['ip_address'] = $ip_address;
            $intConfig['domain_name_server'] = $domain_name_server;
            $intConfig['routers'] = $routers;
            $intConfig['static'] = "true";
            $intConfig['failover'] = "false";
            write_php_ini($intConfig, RASPI_CONFIG_NETWORKING.'/'.$_POST['interface'].".ini");
        }

        $config = join(PHP_EOL, $config);
        file_put_contents("/tmp/dhcpddata", $config);
        system('sudo cp /tmp/dhcpddata '.RASPI_DHCPCD_CONFIG, $return);

        if ($return == 0) {
            $status->addMessage('Wifi Hotspot settings saved', 'success');
        } else {
            $status->addMessage('Unable to save wifi hotspot settings', 'danger');
        }
    } else {
        $status->addMessage('Unable to save wifi hotspot settings', 'danger');
        return false;
    }

    return true;
}
```