⑂ 5a7b774598 ⌄   ...

raspap-webgui / installers / common.sh

billz Update w/ _install_mobile_clients @zbchristian ✓   ⟳ History

👥 11 contributors   👤👤👤👤👤👤👤👤👤👤👤

Executable File │ 682 lines (604 sloc) │ 28.8 KB   ...

```
1    #!/bin/bash
2    #
3    # RaspAP installation functions
4    # Author: @billz <billzimmerman@gmail.com>
5    # Author URI: https://github.com/billz/
6    # License: GNU General Public License v3.0
7    # License URI: https://github.com/raspap/raspap-webgui/blob/master/LICENSE
8
9    # You are not obligated to bundle the LICENSE file with your RaspAP projects as long
10   # as you leave these references intact in the header comments of your source files.
11
12   # Exit on error
13   set -o errexit
14   # Exit on error inside functions
15   set -o errtrace
16   # Turn on traces, disabled by default
17   # set -o xtrace
18
19   # Set defaults
20   readonly raspap_dir="/etc/raspap"
21   readonly raspap_user="www-data"
22   readonly raspap_sudoers="/etc/sudoers.d/090_raspap"
23   readonly raspap_default="/etc/dnsmasq.d/090_raspap.conf"
24   readonly raspap_wlan0="/etc/dnsmasq.d/090_wlan0.conf"
25   readonly raspap_adblock="/etc/dnsmasq.d/090_adblock.conf"
26   readonly raspap_sysctl="/etc/sysctl.d/90_raspap.conf"
27   readonly raspap_network="$raspap_dir/networking/"
28   readonly raspap_router="/etc/lighttpd/conf-available/50-raspap-router.conf"
29   readonly rulesv4="/etc/iptables/rules.v4"
30   readonly notracking_url="https://raw.githubusercontent.com/notracking/hosts-blocklists/master/"
31   webroot_dir="/var/www/html"
32
33   if [ "$insiders" == 1 ]; then
34       repo="RaspAP/raspap-insiders"
35       branch=${RASPAP_INSIDERS_LATEST}
36   fi
37   git_source_url="https://github.com/$repo"
38
39   # NOTE: all the below functions are overloadable for system-specific installs
40   function _install_raspap() {
41       _display_welcome
42       _config_installation
43       _update_system_packages
44       _install_dependencies
45       _enable_php_lighttpd
46       _create_raspap_directories
47       _optimize_php
48       _check_for_old_configs
49       _download_latest_files
50       _change_file_ownership
51       _create_hostapd_scripts
52       _create_lighttpd_scripts
53       _install_lighttpd_configs
54       _move_config_file
55       _default_configuration
56       _configure_networking
57       _prompt_install_adblock
58       _prompt_install_openvpn
59       _install_mobile_clients
60       _prompt_install_wireguard
61       _patch_system_files
62       _install_complete
63   }
64
65   # search for optional installation files names install_feature_*.sh
66   function _install_mobile_clients() {
67       if [ "$insiders" == 1 ]; then
68           _install_log "Installing support for mobile data clients"
69           for feature in $(ls $webroot_dir/installers/install_feature_*.sh) ; do
70               source $feature
71               f=$(basename $feature)
72               func="_${f%.*}"
73               if declare -f -F $func > /dev/null; then
74                   echo "Installing $func"
75                   $func || _install_status 1 "Unable to install feature ($func)"
76               else
77                   _install_status 1 "Install file $f is missing install function $func"
78               fi
```

```bash
79              done
80          fi
81  }
82
83  # Prompts user to set installation options
84  function _config_installation() {
85      if [ "$upgrade" == 1 ]; then
86          opt=(Upgrade Upgrading upgrade)
87      else
88          opt=(Install Installing installation)
89      fi
90      _install_log "Configure ${opt[2]}"
91      _get_linux_distro
92      echo "Detected OS: ${DESC}"
93      echo "Using GitHub repository: ${repo} ${branch} branch"
94      echo "Configuration directory: ${raspap_dir}"
95      echo -n "lighttpd root: ${webroot_dir}? [Y/n]: "
96      if [ "$assume_yes" == 0 ]; then
97          read answer < /dev/tty
98          if [ "$answer" != "${answer#[Nn]}" ]; then
99              read -e -p < /dev/tty "Enter alternate lighttpd directory: " -i "/var/www/html" webroot_dir
100         fi
101     else
102         echo -e
103     fi
104     echo "${opt[1]} lighttpd directory: ${webroot_dir}"
105     if [ "$upgrade" == 1 ]; then
106         echo "This will upgrade your existing install to version ${RASPAP_RELEASE}"
107         echo "Your configuration will NOT be changed"
108     fi
109     echo -n "Complete ${opt[2]} with these values? [Y/n]: "
110     if [ "$assume_yes" == 0 ]; then
111         read answer < /dev/tty
112         if [ "$answer" != "${answer#[Nn]}" ]; then
113             echo "Installation aborted."
114             exit 0
115         fi
116     else
117         echo -e
118     fi
119 }
120
121 # Determines host Linux distribution details
122 function _get_linux_distro() {
123     if type lsb_release >/dev/null 2>&1; then # linuxbase.org
124         OS=$(lsb_release -si)
125         RELEASE=$(lsb_release -sr)
126         CODENAME=$(lsb_release -sc)
127         DESC=$(lsb_release -sd)
128     elif [ -f /etc/os-release ]; then # freedesktop.org
129         . /etc/os-release
130         OS=$ID
131         RELEASE=$VERSION_ID
132         CODENAME=$VERSION_CODENAME
133         DESC=$PRETTY_NAME
134     else
135         _install_status 1 "Unsupported Linux distribution"
136     fi
137 }
138
139 # Sets php package option based on Linux version, abort if unsupported distro
140 function _set_php_package() {
141     case $RELEASE in
142         18.04|19.10) # Ubuntu Server
143             php_package="php7.4-cgi"
144             phpcgiconf="/etc/php/7.4/cgi/php.ini" ;;
145         10*)
146             php_package="php7.3-cgi"
147             phpcgiconf="/etc/php/7.3/cgi/php.ini" ;;
148         9*)
149             php_package="php7.0-cgi"
150             phpcgiconf="/etc/php/7.0/cgi/php.ini" ;;
151         8)
152             _install_status 1 "${DESC} and php5 are not supported. Please upgrade." ;;
153         *)
154             _install_status 1 "${DESC} is unsupported. Please install on a supported distro." ;;
155     esac
156 }
157
158 # Runs a system software update to make sure we're using all fresh packages
159 function _install_dependencies() {
160     _install_log "Installing required packages"
161     _set_php_package
162     if [ "$php_package" = "php7.4-cgi" ]; then
163         echo "Adding apt-repository ppa:ondrej/php"
164         sudo apt-get install $apt_option software-properties-common || _install_status 1 "Unable to install dependency"
165         sudo add-apt-repository $apt_option ppa:ondrej/php || _install_status 1 "Unable to add-apt-repository ppa:ondrej/php"
166     fi
167     if [ ${OS,,} = "debian" ] || [ ${OS,,} = "ubuntu" ]; then
168         dhcpcd_package="dhcpcd5"
169     fi
170     # Set dconf-set-selections
171     echo iptables-persistent iptables-persistent/autosave_v4 boolean true | sudo debconf-set-selections
172     echo iptables-persistent iptables-persistent/autosave_v6 boolean true | sudo debconf-set-selections
173     sudo apt-get install $apt_option lighttpd git hostapd dnsmasq iptables-persistent $php_package $dhcpcd_package vnstat qrencode || _install_status 1 "Unable to install dependen
174     _install_status 0
175 }
176
```

```bash
177    # Enables PHP for lighttpd and restarts service for settings to take effect
178    function _enable_php_lighttpd() {
179        _install_log "Enabling PHP for lighttpd"
180        sudo lighttpd-enable-mod fastcgi-php
181        sudo service lighttpd force-reload
182        sudo systemctl restart lighttpd.service || _install_status 1 "Unable to restart lighttpd"
183    }
184
185    # Verifies existence and permissions of RaspAP directory
186    function _create_raspap_directories() {
187        _install_log "Creating RaspAP directories"
188        if [ -d "$raspap_dir" ]; then
189            sudo mv $raspap_dir "$raspap_dir.`date +%F-%R`" || _install_status 1 "Unable to move old '$raspap_dir' out of the way"
190        fi
191        sudo mkdir -p "$raspap_dir" || _install_status 1 "Unable to create directory '$raspap_dir'"
192
193        # Create a directory for existing file backups.
194        sudo mkdir -p "$raspap_dir/backups"
195
196        # Create a directory to store networking configs
197        echo "Creating $raspap_dir/networking"
198        sudo mkdir -p "$raspap_dir/networking"
199        # Copy existing dhcpcd.conf to use as base config
200        echo "Adding /etc/dhcpcd.conf as base configuration"
201        cat /etc/dhcpcd.conf | sudo tee -a /etc/raspap/networking/defaults > /dev/null
202        echo "Changing file ownership of $raspap_dir"
203        sudo chown -R $raspap_user:$raspap_user "$raspap_dir" || _install_status 1 "Unable to change file ownership for '$raspap_dir'"
204    }
205
206    # Generate hostapd logging and service control scripts
207    function _create_hostapd_scripts() {
208        _install_log "Creating hostapd logging & control scripts"
209        sudo mkdir $raspap_dir/hostapd || _install_status 1 "Unable to create directory '$raspap_dir/hostapd'"
210
211        # Move logging shell scripts
212        sudo cp "$webroot_dir/installers/"*log.sh "$raspap_dir/hostapd" || _install_status 1 "Unable to move logging scripts"
213        # Move service control shell scripts
214        sudo cp "$webroot_dir/installers/"service*.sh "$raspap_dir/hostapd" || _install_status 1 "Unable to move service control scripts"
215        # Make enablelog.sh and disablelog.sh not writable by www-data group.
216        sudo chown -c root:"$raspap_user" "$raspap_dir/hostapd/"*.sh || _install_status 1 "Unable change owner and/or group"
217        sudo chmod 750 "$raspap_dir/hostapd/"*.sh || _install_status 1 "Unable to change file permissions"
218        _install_status 0
219    }
220
221    # Generate lighttpd service control scripts
222    function _create_lighttpd_scripts() {
223        _install_log "Creating lighttpd control scripts"
224        sudo mkdir $raspap_dir/lighttpd || _install_status 1 "Unable to create directory '$raspap_dir/lighttpd'"
225
226        # Move service control shell scripts
227        echo "Copying configport.sh to $raspap_dir/lighttpd"
228        sudo cp "$webroot_dir/installers/"configport.sh "$raspap_dir/lighttpd" || _install_status 1 "Unable to move service control scripts"
229        # Make configport.sh writable by www-data group.
230        echo "Changing file ownership"
231        sudo chown -c root:"$raspap_user" "$raspap_dir/lighttpd/"*.sh || _install_status 1 "Unable change owner and/or group"
232        sudo chmod 750 "$raspap_dir/lighttpd/"*.sh || _install_status 1 "Unable to change file permissions"
233        _install_status 0
234    }
235
236    # Copy extra config files required to configure lighttpd
237    function _install_lighttpd_configs() {
238        _install_log "Copying lighttpd extra config files"
239
240        # Copy config files
241        echo "Copying 50-raspap-router.conf to /etc/lighttpd/conf-available"
242
243        CONFSRC="$webroot_dir/config/50-raspap-router.conf"
244        LTROOT=$(grep "server.document-root" /etc/lighttpd/lighttpd.conf | awk -F '=' '{print $2}' | tr -d " \"")
245
246        # Compare values and get difference
247        HTROOT=${webroot_dir/$LTROOT}
248
249        # Remove trailing slash if present
250        HTROOT=$(echo "$HTROOT" | sed -e 's/\/$//')
251
252        # Substitute values
253        awk "{gsub(\"/REPLACE_ME\",\"$HTROOT\")}1" $CONFSRC > /tmp/50-raspap-router.conf
254
255        # Copy into place
256        sudo cp /tmp/50-raspap-router.conf /etc/lighttpd/conf-available/ || _install_status 1 "Unable to copy lighttpd config file into place."
257
258        # Link into conf-enabled
259        echo "Creating link to /etc/lighttpd/conf-enabled"
260        if ! [ -L $raspap_router ]; then
261            echo "Existing 50-raspap-router.conf found. Unlinking."
262            sudo unlink "/etc/lighttpd/conf-enabled/50-raspap-router.conf"
263        fi
264        echo "Linking 50-raspap-router.conf to /etc/lighttpd/conf-enabled/"
265        sudo ln -s "/etc/lighttpd/conf-available/50-raspap-router.conf" "/etc/lighttpd/conf-enabled/50-raspap-router.conf" || _install_status 1 "Unable to symlink lighttpd config file
266        sudo systemctl restart lighttpd.service || _install_status 1 "Unable to restart lighttpd"
267        _install_status 0
268    }
269
270
271    # Prompt to install ad blocking
272    function _prompt_install_adblock() {
273        _install_log "Configure ad blocking (Beta)"
274        echo -n "Install ad blocking and enable list management? [Y/n]: "
```

```
275          if [ "$assume_yes" == 0 ]; then
276              read answer < /dev/tty
277              if [ "$answer" != "${answer#[Nn]}" ]; then
278                  echo -e
279              else
280                  _install_adblock
281              fi
282          elif [ "$adblock_option" == 1 ]; then
283              _install_adblock
284          else
285              echo "(Skipped)"
286          fi
287      }
288
289      # Download notracking adblock lists and enable option
290      function _install_adblock() {
291          _install_log "Creating ad blocking base configuration (Beta)"
292          if [ ! -d "$raspap_dir/adblock" ]; then
293              echo "Creating $raspap_dir/adblock"
294              sudo mkdir -p "$raspap_dir/adblock"
295          fi
296          if [ ! -f /tmp/hostnames.txt ]; then
297              echo "Fetching latest hostnames list"
298              wget ${notracking_url}hostnames.txt -q --show-progress --progress=bar:force -O /tmp/hostnames.txt 2>&1 \
299                  || _install_status 1 "Unable to download notracking hostnames"
300          fi
301          if [ ! -f /tmp/domains.txt ]; then
302              echo "Fetching latest domains list"
303              wget ${notracking_url}domains.txt -q --show-progress --progress=bar:force -O /tmp/domains.txt 2>&1 \
304                  || _install_status 1 "Unable to download notracking domains"
305          fi
306          echo "Adding blocklists to $raspap_dir/adblock"
307          sudo cp /tmp/hostnames.txt $raspap_dir/adblock || _install_status 1 "Unable to move notracking hostnames"
308          sudo cp /tmp/domains.txt $raspap_dir/adblock || _install_status 1 "Unable to move notracking domains"
309
310          echo "Moving and setting permissions for blocklist update script"
311          sudo cp "$webroot_dir/installers/"update_blocklist.sh $raspap_dir/adblock" || _install_status 1 "Unable to move blocklist update script"
312
313          # Make blocklists and update script writable by www-data group
314          sudo chown -c root:"$raspap_user" "$raspap_dir/adblock/"*.* || _install_status 1 "Unable to change owner/group"
315          sudo chmod 750 "$raspap_dir/adblock/"*.sh || install_error "Unable to change file permissions"
316
317          # Create 090_adblock.conf and write values to /etc/dnsmasq.d
318          if [ ! -f "$raspap_adblock" ]; then
319              echo "Adding 090_addblock.conf to /etc/dnsmasq.d"
320              sudo touch "$raspap_adblock"
321              echo "conf-file=$raspap_dir/adblock/domains.txt" | sudo tee -a "$raspap_adblock" > /dev/null || _install_status 1 "Unable to write to $raspap_adblock"
322              echo "addn-hosts=$raspap_dir/adblock/hostnames.txt" | sudo tee -a "$raspap_adblock" > /dev/null || _install_status 1 "Unable to write to $raspap_adblock"
323          fi
324
325          # Remove dhcp-option=6 in dnsmasq.d/090_wlan0.conf to force local DNS resolution for DHCP clients
326          echo "Enabling local DNS name resolution for DHCP clients"
327          sudo sed -i '/dhcp-option=6/d' $raspap_wlan0 || _install_status 1 "Unable to modify $raspap_dnsmasq"
328
329          echo "Enabling ad blocking management option"
330          sudo sed -i "s/\('RASPI_ADBLOCK_ENABLED', \)false/\1true/g" "$webroot_dir/includes/config.php" || _install_status 1 "Unable to modify config.php"
331          _install_status 0
332      }
333
334      # Prompt to install openvpn
335      function _prompt_install_openvpn() {
336          _install_log "Configure OpenVPN support"
337          echo -n "Install OpenVPN and enable client configuration? [Y/n]: "
338          if [ "$assume_yes" == 0 ]; then
339              read answer < /dev/tty
340              if [ "$answer" != "${answer#[Nn]}" ]; then
341                  echo -e
342              else
343                  _install_openvpn
344              fi
345          elif [ "$ovpn_option" == 1 ]; then
346              _install_openvpn
347          else
348              echo "(Skipped)"
349          fi
350      }
351
352      # Prompt to install WireGuard
353      function _prompt_install_wireguard() {
354          if [ "$insiders" == 1 ]; then
355              _install_log "Configure WireGuard support"
356              echo -n "Install WireGuard and enable VPN tunnel configuration? [Y/n]: "
357              if [ "$assume_yes" == 0 ]; then
358                  read answer < /dev/tty
359                  if [ "$answer" != "${answer#[Nn]}" ]; then
360                      echo -e
361                  else
362                      _install_wireguard
363                  fi
364              elif [ "$wg_option" == 1 ]; then
365                  _install_wireguard
366              else
367                  echo "(Skipped)"
368              fi
369          fi
370      }
371
372      # Install Wireguard from the Debian unstable distro
```

```
373   function _install_wireguard() {
374       _install_log "Configure WireGuard support"
375       if [ "$OS" == "Debian" ]; then
376           echo 'deb http://ftp.debian.org/debian buster-backports main' | sudo tee /etc/apt/sources.list.d/buster-backports.list || _install_status 1 "Unable to add Debian backports
377       fi
378       echo "Installing wireguard from apt"
379       sudo apt-get update && sudo apt-get install $apt_option wireguard || _install_status 1 "Unable to install wireguard"
380       echo "Enabling wg-quick@wg0"
381       sudo systemctl enable wg-quick@wg0 || _install_status 1 "Failed to enable wg-quick service"
382       echo "Enabling WireGuard management option"
383       sudo sed -i "s/\(\'RASPI_WIREGUARD_ENABLED', \)false/\1true/g" "$webroot_dir/includes/config.php" || _install_status 1 "Unable to modify config.php"
384       _install_status 0
385   }
386
387   # Install openvpn and enable client configuration option
388   function _install_openvpn() {
389       _install_log "Installing OpenVPN and enabling client configuration"
390       echo "Adding packages via apt-get"
391       sudo apt-get install -y openvpn || _install_status 1 "Unable to install openvpn"
392       sudo sed -i "s/\(\'RASPI_OPENVPN_ENABLED', \)false/\1true/g" "$webroot_dir/includes/config.php" || _install_status 1 "Unable to modify config.php"
393       echo "Enabling openvpn-client service on boot"
394       sudo systemctl enable openvpn-client@client || _install_status 1 "Unable to enable openvpn-client daemon"
395       _create_openvpn_scripts || _install_status 1 "Unable to create openvpn control scripts"
396   }
397
398   # Generate openvpn logging and auth control scripts
399   function _create_openvpn_scripts() {
400       _install_log "Creating OpenVPN control scripts"
401       sudo mkdir $raspap_dir/openvpn || _install_status 1 "Unable to create directory '$raspap_dir/openvpn'"
402
403       # Move service auth control & logging shell scripts
404       sudo cp "$webroot_dir/installers/"configauth.sh "$raspap_dir/openvpn" || _install_status 1 "Unable to move auth control script"
405       sudo cp "$webroot_dir/installers/"openvpnlog.sh "$raspap_dir/openvpn" || _install_status 1 "Unable to move logging script"
406       # Make scripts executable by www-data group
407       sudo chown -c root:"$raspap_user" "$raspap_dir/openvpn/"*.sh || _install_status 1 "Unable change owner and/or group"
408       sudo chmod 750 "$raspap_dir/openvpn/"*.sh || _install_status 1 "Unable to change file permissions"
409       _install_status 0
410   }
411
412   # Fetches latest files from github to webroot
413   function _download_latest_files() {
414       if [ ! -d "$webroot_dir" ]; then
415           sudo mkdir -p $webroot_dir || _install_status 1 "Unable to create new webroot directory"
416       fi
417
418       if [ -d "$webroot_dir" ]; then
419           sudo mv $webroot_dir "$webroot_dir.`date +%F-%R`" || _install_status 1 "Unable to remove old webroot directory"
420       fi
421
422       _install_log "Cloning latest files from github"
423       git clone --branch $branch --depth 1 -c advice.detachedHead=false $git_source_url /tmp/raspap-webgui || _install_status 1 "Unable to download files from github"
424
425       sudo mv /tmp/raspap-webgui $webroot_dir || _install_status 1 "Unable to move raspap-webgui to web root"
426       if [ "$upgrade" == 1 ]; then
427           _install_log "Applying existing configuration to ${webroot_dir}/includes"
428           sudo mv /tmp/config.php $webroot_dir/includes || _install_status 1 "Unable to move config.php to ${webroot_dir}/includes"
429       fi
430
431       _install_status 0
432   }
433
434   # Sets files ownership in web root directory
435   function _change_file_ownership() {
436       if [ ! -d "$webroot_dir" ]; then
437           _install_status 1 "Web root directory doesn't exist"
438       fi
439
440       _install_log "Changing file ownership in web root directory"
441       sudo chown -R $raspap_user:$raspap_user "$webroot_dir" || _install_status 1 "Unable to change file ownership for '$webroot_dir'"
442   }
443
444   # Check for existing configuration files
445   function _check_for_old_configs() {
446       if [ "$upgrade" == 1 ]; then
447           _install_log "Moving existing configuration to /tmp"
448           sudo mv $webroot_dir/includes/config.php /tmp || _install_status 1 "Unable to move config.php to /tmp"
449       else
450           _install_log "Backing up existing configs to ${raspap_dir}/backups"
451           if [ -f /etc/network/interfaces ]; then
452               sudo cp /etc/network/interfaces "$raspap_dir/backups/interfaces.`date +%F-%R`"
453               sudo ln -sf "$raspap_dir/backups/interfaces.`date +%F-%R`" "$raspap_dir/backups/interfaces"
454           fi
455
456           if [ -f /etc/hostapd/hostapd.conf ]; then
457               sudo cp /etc/hostapd/hostapd.conf "$raspap_dir/backups/hostapd.conf.`date +%F-%R`"
458               sudo ln -sf "$raspap_dir/backups/hostapd.conf.`date +%F-%R`" "$raspap_dir/backups/hostapd.conf"
459           fi
460
461           if [ -f $raspap_default ]; then
462               sudo cp $raspap_default "$raspap_dir/backups/090_raspap.conf.`date +%F-%R`"
463               sudo ln -sf "$raspap_dir/backups/090_raspap.conf.`date +%F-%R`" "$raspap_dir/backups/090_raspap.conf"
464           fi
465
466           if [ -f $raspap_wlan0 ]; then
467               sudo cp $raspap_wlan0 "$raspap_dir/backups/090_wlan0.conf.`date +%F-%R`"
468               sudo ln -sf "$raspap_dir/backups/090_wlan0.conf.`date +%F-%R`" "$raspap_dir/backups/090_wlan0.conf"
469           fi
470
```

```bash
471         if [ -f /etc/dhcpcd.conf ]; then
472             sudo cp /etc/dhcpcd.conf "$raspap_dir/backups/dhcpcd.conf.`date +%F-%R`"
473             sudo ln -sf "$raspap_dir/backups/dhcpcd.conf.`date +%F-%R`" "$raspap_dir/backups/dhcpcd.conf"
474         fi
475
476         for file in /etc/systemd/network/raspap-*.net*; do
477             if [ -f "${file}" ]; then
478                 filename=$(basename $file)
479                 sudo cp "$file" "${raspap_dir}/backups/${filename}.`date +%F-%R`"
480                 sudo ln -sf "${raspap_dir}/backups/${filename}.`date +%F-%R`" "${raspap_dir}/backups/${filename}"
481             fi
482         done
483     fi
484     _install_status 0
485 }
486
487 # Move configuration file to the correct location
488 function _move_config_file() {
489     if [ ! -d "$raspap_dir" ]; then
490         _install_status 1 "'$raspap_dir' directory doesn't exist"
491     fi
492
493     _install_log "Moving configuration file to $raspap_dir"
494     sudo cp "$webroot_dir"/raspap.php "$raspap_dir" || _install_status 1 "Unable to move files to '$raspap_dir'"
495     sudo chown -R $raspap_user:$raspap_user "$raspap_dir" || _install_status 1 "Unable to change file ownership for '$raspap_dir'"
496 }
497
498 # Set up default configuration
499 function _default_configuration() {
500     if [ "$upgrade" == 0 ]; then
501         _install_log "Applying default configuration to installed services"
502
503         sudo cp $webroot_dir/config/hostapd.conf /etc/hostapd/hostapd.conf || _install_status 1 "Unable to move hostapd configuration file"
504         sudo cp $webroot_dir/config/090_raspap.conf $raspap_default || _install_status 1 "Unable to move dnsmasq default configuration file"
505         sudo cp $webroot_dir/config/090_wlan0.conf $raspap_wlan0 || _install_status 1 "Unable to move dnsmasq wlan0 configuration file"
506         sudo cp $webroot_dir/config/dhcpcd.conf /etc/dhcpcd.conf || _install_status 1 "Unable to move dhcpcd configuration file"
507         sudo cp $webroot_dir/config/defaults.json $raspap_network || _install_status 1 "Unable to move defaults.json settings"
508
509         echo "Changing file ownership of $raspap_dir"
510         sudo chown -R $raspap_user:$raspap_user "$raspap_dir" || _install_status 1 "Unable to change file ownership for '$raspap_dir'"
511
512         echo "Checking for existence of /etc/dnsmasq.d"
513         [ -d /etc/dnsmasq.d ] || sudo mkdir /etc/dnsmasq.d
514
515         echo "Copying bridged AP config to /etc/systemd/network"
516         sudo systemctl stop systemd-networkd
517         sudo systemctl disable systemd-networkd
518         sudo cp $webroot_dir/config/raspap-bridge-br0.netdev /etc/systemd/network/raspap-bridge-br0.netdev || _install_status 1 "Unable to move br0 netdev file"
519         sudo cp $webroot_dir/config/raspap-br0-member-eth0.network /etc/systemd/network/raspap-br0-member-eth0.network || _install_status 1 "Unable to move br0 member file"
520
521         echo "Copying primary RaspAP config to includes/config.php"
522         if [ ! -f "$webroot_dir/includes/config.php" ]; then
523             sudo cp "$webroot_dir/config/config.php" "$webroot_dir/includes/config.php"
524         fi
525         _install_status 0
526     fi
527 }
528
529 # Install and enable RaspAP daemon
530 function _enable_raspap_daemon() {
531     _install_log "Enabling RaspAP daemon"
532     echo "Disable with: sudo systemctl disable raspapd.service"
533     sudo cp $webroot_dir/installers/raspapd.service /lib/systemd/system/ || _install_status 1 "Unable to move raspap.service file"
534     sudo systemctl daemon-reload
535     sudo systemctl enable raspapd.service || _install_status 1 "Failed to enable raspap.service"
536 }
537
538 # Configure IP forwarding, set IP tables rules, prompt to install RaspAP daemon
539 function _configure_networking() {
540     _install_log "Configuring networking"
541     echo "Enabling IP forwarding"
542     echo "net.ipv4.ip_forward=1" | sudo tee $raspap_sysctl > /dev/null || _install_status 1 "Unable to set IP forwarding"
543     sudo sysctl -p $raspap_sysctl || _install_status 1 "Unable to execute sysctl"
544     sudo /etc/init.d/procps restart || _install_status 1 "Unable to execute procps"
545
546     echo "Checking iptables rules"
547     rules=(
548     "-A POSTROUTING -j MASQUERADE"
549     "-A POSTROUTING -s 192.168.50.0/24 ! -d 192.168.50.0/24 -j MASQUERADE"
550     )
551     for rule in "${rules[@]}"; do
552         if grep -- "$rule" $rulesv4 > /dev/null; then
553             echo "Rule already exits: ${rule}"
554         else
555             rule=$(sed -e 's/^\(-A POSTROUTING\)/-t nat \1/' <<< $rule)
556             echo "Adding rule: ${rule}"
557             sudo iptables $rule || _install_status 1 "Unable to execute iptables"
558             added=true
559         fi
560     done
561     # Persist rules if added
562     if [ "$added" = true ]; then
563         echo "Persisting IP tables rules"
564         sudo iptables-save | sudo tee $rulesv4 > /dev/null || _install_status 1 "Unable to execute iptables-save"
565     fi
566
567     # Prompt to install RaspAP daemon
568     echo -n "Enable RaspAP control service (Recommended)? [Y/n]: "
```

```
569        if [ "$assume_yes" == 0 ]; then
570            read answer < /dev/tty
571            if [ "$answer" != "${answer#[Nn]}" ]; then
572                echo -e
573            else
574                _enable_raspap_daemon
575            fi
576        else
577            echo -e
578            _enable_raspap_daemon
579        fi
580        _install_status 0
581   }
582
583   # Add sudoers file to /etc/sudoers.d/ and set file permissions
584   function _patch_system_files() {
585
586        # Create sudoers if not present
587        if [ ! -f $raspap_sudoers ]; then
588            _install_log "Adding raspap.sudoers to ${raspap_sudoers}"
589            sudo cp "$webroot_dir/installers/raspap.sudoers" $raspap_sudoers || _install_status 1 "Unable to apply raspap.sudoers to $raspap_sudoers"
590            sudo chmod 0440 $raspap_sudoers || _install_status 1 "Unable to change file permissions for $raspap_sudoers"
591        fi
592
593        # Add symlink to prevent wpa_cli cmds from breaking with multiple wlan interfaces
594        _install_log "Symlinked wpa_supplicant hooks for multiple wlan interfaces"
595        if [ ! -f /usr/share/dhcpcd/hooks/10-wpa_supplicant ]; then
596            sudo ln -s /usr/share/dhcpcd/hooks/10-wpa_supplicant /etc/dhcp/dhclient-enter-hooks.d/
597        fi
598
599        # Unmask and enable hostapd.service
600        _install_log "Unmasking and enabling hostapd service"
601        sudo systemctl unmask hostapd.service
602        sudo systemctl enable hostapd.service
603        _install_status 0
604   }
605
606
607   # Optimize configuration of php-cgi.
608   function _optimize_php() {
609        if [ "$upgrade" == 0 ]; then
610            _install_log "Optimize PHP configuration"
611            if [ ! -f "$phpcgiconf" ]; then
612                _install_warning "PHP configuration could not be found."
613                return
614            fi
615
616            # Backup php.ini and create symlink for restoring.
617            datetimephpconf=$(date +%F-%R)
618            sudo cp "$phpcgiconf" "$raspap_dir/backups/php.ini.$datetimephpconf"
619            sudo ln -sf "$raspap_dir/backups/php.ini.$datetimephpconf" "$raspap_dir/backups/php.ini"
620
621            echo -n "Enable HttpOnly for session cookies (Recommended)? [Y/n]: "
622            if [ "$assume_yes" == 0 ]; then
623                read answer < /dev/tty
624                if [ "$answer" != "${answer#[Nn]}" ]; then
625                    echo -e
626                else
627                    php_session_cookie=1;
628                fi
629            fi
630
631            if [ "$assume_yes" == 1 ] || [ "$php_session_cookie" == 1 ]; then
632                echo "Php-cgi enabling session.cookie_httponly."
633                sudo sed -i -E 's/^session\.cookie_httponly\s*=\s*(0|([Oo]ff)|([Ff]alse)|([Nn]o))\s*$/session.cookie_httponly = 1/' "$phpcgiconf"
634            fi
635
636            if [ "$php_package" = "php7.1-cgi" ]; then
637                echo -n "Enable PHP OPCache (Recommended)? [Y/n]: "
638                if [ "$assume_yes" == 0 ]; then
639                    read answer < /dev/tty
640                    if [ "$answer" != "${answer#[Nn]}" ]; then
641                        echo -e
642                    else
643                        php_opcache=1;
644                    fi
645                fi
646
647                if [ "$assume_yes" == 1 ] || [ "$phpopcache" == 1 ]; then
648                    echo -e "Php-cgi enabling opcache.enable."
649                    sudo sed -i -E 's/^;?opcache\.enable\s*=\s*(0|([Oo]ff)|([Ff]alse)|([Nn]o))\s*$/opcache.enable = 1/' "$phpcgiconf"
650                    # Make sure opcache extension is turned on.
651                    if [ -f "/usr/sbin/phpenmod" ]; then
652                        sudo phpenmod opcache
653                    else
654                        _install_status 2 "phpenmod not found."
655                    fi
656                fi
657            fi
658        fi
659   }
660
661   function _install_complete() {
662        _install_log "Installation completed"
663        echo "Join RaspAP Insiders for early access to exclusive features!"
664        echo -e "${ANSI_RASPBERRY}"
665        echo "> https://docs.raspap.com/insiders/"
666        echo "> https://github.com/sponsors/RaspAP/"
```

```
667        echo -e "${ANSI_RESET}"
668    if [ "$assume_yes" == 0 ]; then
669        # Prompt to reboot if wired ethernet (eth0) is connected.
670        # With default_configuration this will create an active AP on restart.
671        if ip a | grep -q ': eth0:.*state UP'; then
672            echo -n "The system needs to be rebooted as a final step. Reboot now? [y/N]: "
673            read answer < /dev/tty
674            if [ "$answer" != "${answer#[Nn]}" ]; then
675                echo "Installation reboot aborted."
676                exit 0
677            fi
678            sudo shutdown -r now || _install_status 1 "Unable to execute shutdown"
679        fi
680    fi
681 }
682
```