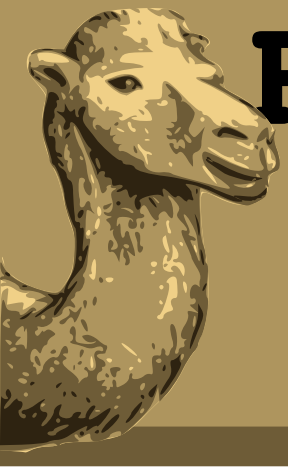
Perl Mongers im Ruhrgebiet



Ruhr.pm

A No-Frills Zero-Management Visitors' WLAN
for less than 30 Euros

Autor: Veit Wahlich

EMail: veit AT ruhr.pm.org

Datum: 10. Mai 2010

http://ruhr.pm.org/

Ruhr.pm

Application Purpose

- simple, yet secure Wi-Fi internet access for your
  - restaurant/bar/cafe/lounge customers
  - office visitors
  - party guests
- preventing your wireless key gets around your neighbourhood without changing it periodically by hand

Ruhr.pm

No-Frills

- beware using compatibility-critical network access control such as
  - IEEE 802.1X
  - captive portals
- avoid requiring any non-standard application such as
  - proprietary authentication software
  - VPN clients

Ruhr.pm

Zero-Management

- must run independently
  - no operation of a RADIUS server
  - no management of a CA
  - no requirement of a controlling terminal
- must not require direct interaction
  - no adding or registering of users
  - no creation and distribution of client or CA certificate files

Ruhr.pm

Low-Cost

- a stand-alone solution employing customer-class Wi-Fi routers running a custom Linux firmware
- applying to the DD-WRT firmware
  - available open source and free of charge
  - has been successfully tested on a huge number of customer-class router devices
  - offers adequate interfaces to be easily extended
    - even without accessing a telnet/ssh console
  - porting to many other Linux-based firmwares without any or with few changes

Ruhr.pm

Low-Cost

- to date it has already been successfully implemented on these low-cost devices:

Vendor && Model	CPU/SoC	ROM/RAM	Antenna	Price
Buffalo WHR-G125	Broadcom 5354	4MB/16MB	R-SMA / -	EUR 24
D-Link DIR-300 A1	Atheros 2317	4MB/16MB	R-SMA	-
D-Link DIR-300 B1	Ralink RT3050	4MB/32MB	R-SMA	EUR 26
Asus WL-500gP v1	Broadcom 4712/4704	8MB/32MB	R-SMA	-
Asus WL-500gP v2	Broadcom 5354	8MB/32MB	R-SMA	EUR 67
Linksys WRT54G v3.1	Broadcom 4712	4MB/16MB	R-TNC	-
Linksys WRT54GL v1.1	Broadcom 5352	4MB/16MB	R-TNC	EUR 46

- prices in Germany as of May 2010, incl. 19% VAT

Ruhr.pm

Resolution Approach

- use a commonly supported wireless security with shared key (PSK)
  - WPA/TKIP for best compatibility
  - WPA2/AES resp. WPA2/CCMP for best security
- create PSKs depending on a secret passphrase and current date or time
  - PSK changing automatically every hour, day, week, month, ...
  - current PSK ascertainable by knowledge of the secret passphrase and date/time format

Ruhr.pm

Resolution Approach

- using cryptographic hashing algorithms
  - current PSK can be communicated to your visitors/customers/guests without risk of deduction of the secret passphrase
  - digest can be shortened to reflect security needs and validity duration
- multiple virtual access points
  - BSSID hardware for multiple APs with own MACs
  - i.e. have one WLAN changing PSK daily, another changing weekly and a third changing never

Ruhr.pm

Resolution Approach

- ascertainable PSKs allow creating a list of dates and associated keys
  - treasure a printed list of the keys for the next months or years
  - no need to calculate or readout keys every day
- DD-WRT brings everything needed for a basic implementation
  - only a few hundred bytes of either NVRAM or JFFS2 storage required
  - even “micro” needs no further package installation

Ruhr.pm

Abstract

- have a key template involving a complex passphrase and a strftime() format template matching the intended validity duration
  - template := “y%Yd%j movezigforgreatjustice\n”
    - substitution characters (amongst others): %Y := year (i.e. “2010”), %j := day of year (i.e. Jan. 1 is “001”), %W := week of year (i.e. “00” for the first week of the year)
    - for security reasons, put the passphrase at the **end** of the template

Ruhr.pm

Abstract

- parse key template and current time through strftime()
  - key := strftime(template, time)
    - time is May 10 2010 here
  - key == “y2010d130 movezigforgreatjustice\n”
- create a cryptographic hash using MD5 algorithm
  - digest := md5(key)
    - md5() has hexadecimal output here
  - digest == “306a531aef1945864f3817cfb02f114c”

Ruhr.pm

Abstract

- shorten the digest to the designated length
  - psk := substr(digest, 0, length)
    - length is 10 here
  - psk == “306a531aef”
- make the PSK better readable/communicatable by replacing the digits 0..9 by the letters g..p
  - psk := ~y/[0-9]/[g-p]/
  - psk == “jgnamjhaef”
- use the PSK as shared secret for your WLAN

Ruhr.pm

Implementation

```
#!/bin/sh

UPD=0

while [ -n "$1" -a -n "$2" -a -n "$3" ]; do

  PSK="$(date "+$2" | md5sum - \
         | sed "s/^\(\w\{$3\}\).*/\1/;
                y/0123456789/ghijkmnopq/")"

  if [ "$PSK"!="$(nvram get "$1_wpa_psk")" ]
  then
    nvram set "$1_wpa_psk=$PSK"
    UPD=1
  fi

  shift 3

done

if [ "$UPD" -gt 0 ]; then
  killall -9 nas
  sleep 1
  /sbin/check_ps
fi
```

This script implements the abstract above in Shell/Busybox, accepting one or more tuples of three command line parameters:
1. the interface to set the PSK for
2. the key template to generate from
3. the length of the PSK to create

Verified changes are written to NVRAM. If an update happened, the network authentication service is being restarted to adopt the changes from NVRAM.

Find an extended version attached.

Ruhr.pm

Security Considerations

- Busybox's md5sum utility solely offers hexadecimal digest output
  - this is limiting the key strength to 16^<length> unless adding further processing
  - Base64 output would be nice, offering 64^<length> key variations
    - i.e. by converting hex to byte and using uuencode -m
    - allowing much shorter keys for the same security, but much more complicated to read and communicate

Ruhr.pm

Security Considerations

- customer-class devices usually do not feature a battery-backed RTC
  - using a NTP, time/udp or time/tcp server required
  - security falls with forgery of or control over the time server(s) used
- the MD5 algorithm has been weakened badly
  - no stronger cryptographic hashing algorithms available in the Busybox used in DD-WRT
  - however colliding hashes does merely marginally affect security in this case

Ruhr.pm

Legal Considerations

- logging requirements
  - in some countries or federal states you may need to track who is using your internet access point during which time
  - much harder to determine which users currently use your AP than on user-based systems
  - consider the legal rules for offering internet access to individuals when implementing this solution (without further precautions)

Ruhr.pm

Setup

- install DD-WRT on a device and configure it according to your needs
  - DD-WRT version used here: v24-preSP2-13064
    - build has a bug when editing existing cronjobs, consider notice stated below
  - in this example, we will create a virtual AP for your visitors, which will cause problems with i.e. Windows OS if your device is not capable to use separate MAC addresses (BSSID)
    - if not sure, see DD-WRT wiki on how to determine

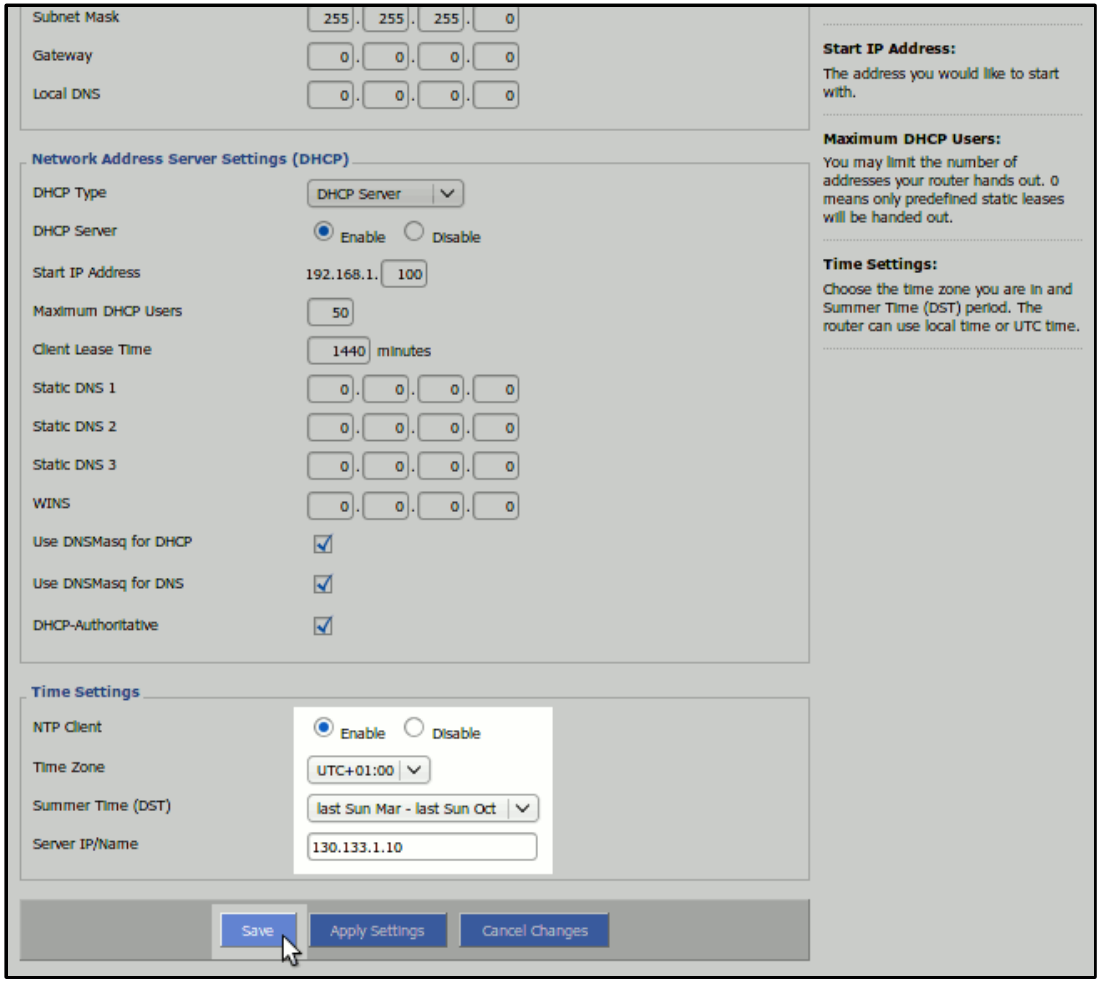
Ruhr.pm

Setup

- as you will provide a time-/date-dependent service, you will require a credible time source
  - on the *Setup* tab, select the *Basic* section and scroll down to *Time Settings*
  - enable the NTP client, select your local time zone and daylight savings directive, enter a **credible** NTP time server and hit *Save*
    - use IP address instead of hostname for the time server to prevent DNS forgery
    - consider running your own local time server

Ruhr.pm

Setup



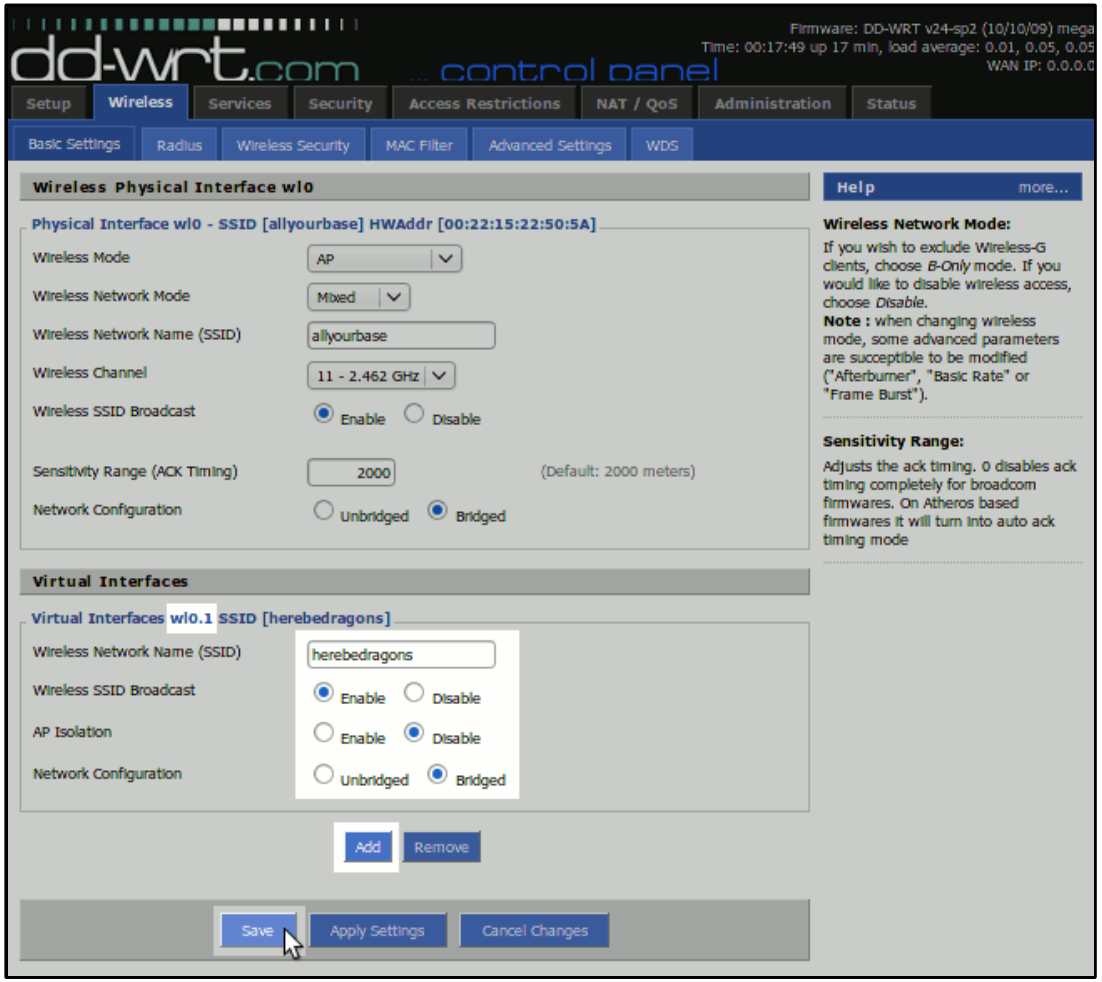
Ruhr.pm

Setup

- add a new virtual access point
  - on the *Wireless* tab navigate to the *Basic Settings* section
  - at *Virtual Interfaces*, hit the *Add* button
  - enter a SSID for your visitors WLAN
  - unless you plan to separate the networks by OSI layer 3 (not discussed here), you want a *Bridged* network configuration and hit *Save*
  - keep the new virtual interface's network device name in mind for later use

Ruhr.pm

Setup



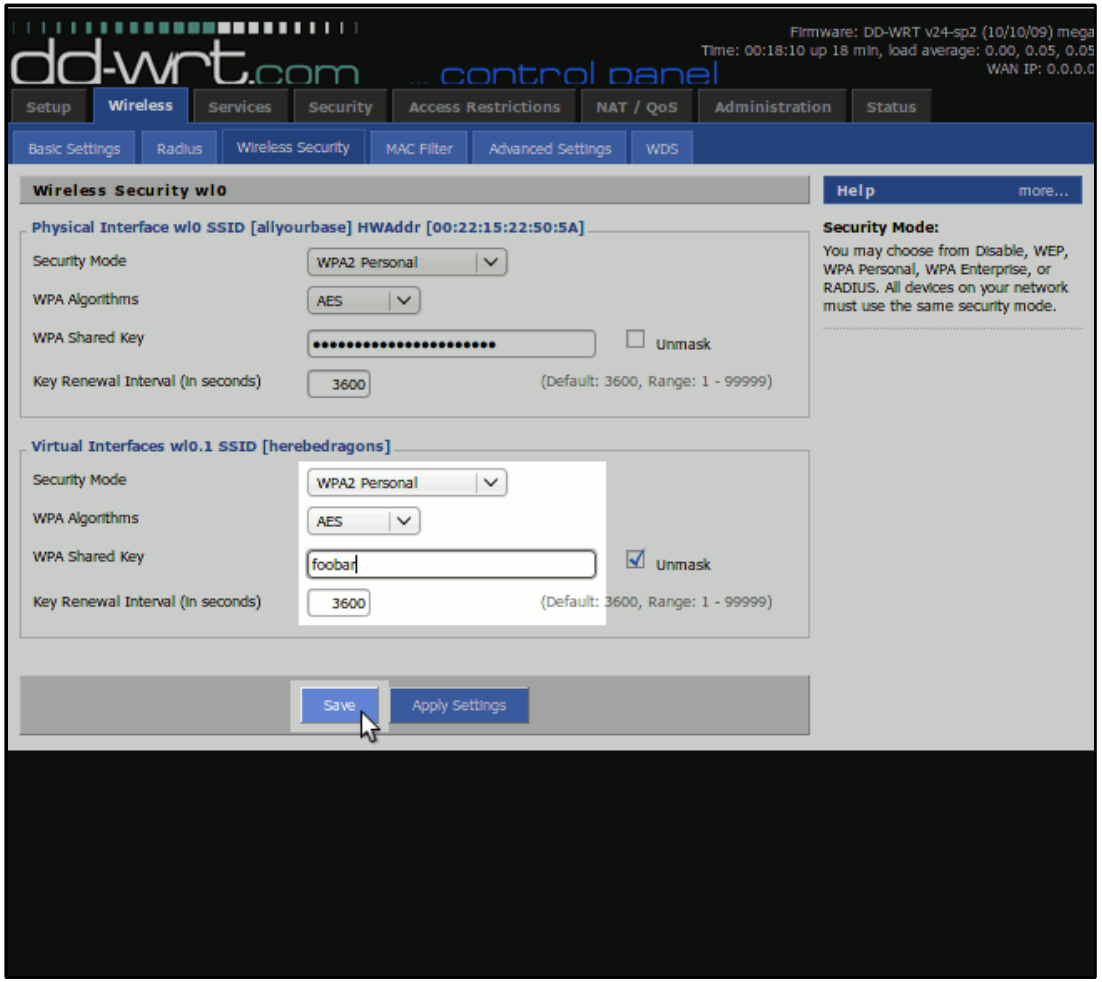
Ruhr.pm

Setup

- configure appropriate network security
  - navigate to the *Wireless Security* section
  - select an appropriate *Security Mode* and *WPA Algorithm* for your visitors' WLAN
    - any WPA/WPA2 personal mode will suffice
    - use WPA/TKIP for best compatibility or WPA2/AES for best security
  - enter any *WPA Shared Key* as placeholder and hit *Save*

Ruhr.pm

Setup



Ruhr.pm

Setup

- configure the router to create the script shown before on its RAM disk at boot time
  - go to the *Administration* tab, select *Commands* and enter the *Command Shell* text area
  - use a single quoted HERE document to create the file as /tmp/update_psk
    - you may want to use the slightly extended version provided with this document and shown in the following screenshot
  - chmod the file to make it executable
  - hit the *Save Startup* button to store it in NVRAM

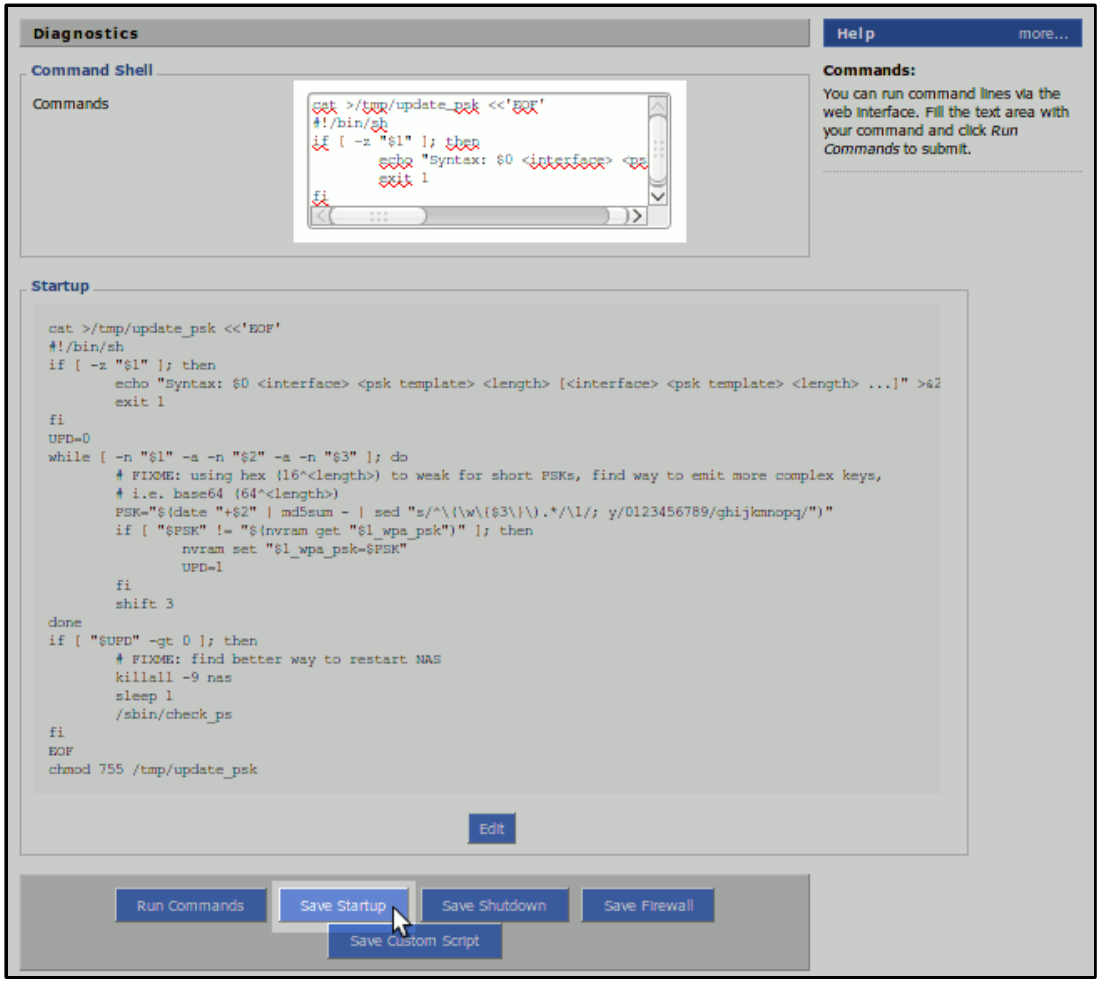
Ruhr.pm

Setup

- configure the router to create the script shown before on its RAM disk at boot time
  - instead of creating the file on every start up of the router, you may want to skip this and store the file in the flash ROM using JFFS

Ruhr.pm

Setup



Ruhr.pm

Setup

- finally create a cronjob that executes the script
  - select the *Administration* tab and navigate to *Management*
  - enable Cron and add a cronjob to the text area
    - run the script every few minutes, even every minute is okay, run it as user root
    - the script takes 3 arguments:
      - the device name kept in mind before
      - the PSK template (a newline is appended automatically)
      - the desired PSK length

Ruhr.pm

Setup

- finally create a cronjob that executes the script
  - hit the *Apply Settings* button to save the cronjob and apply all changes
  - wait some seconds until the device is ready again
  - **Note**: The script can calculate and update the PSK for more than one virtual interface with other key templates and PSK length. Simply add another 3 parameters for your second visitors' WLAN.

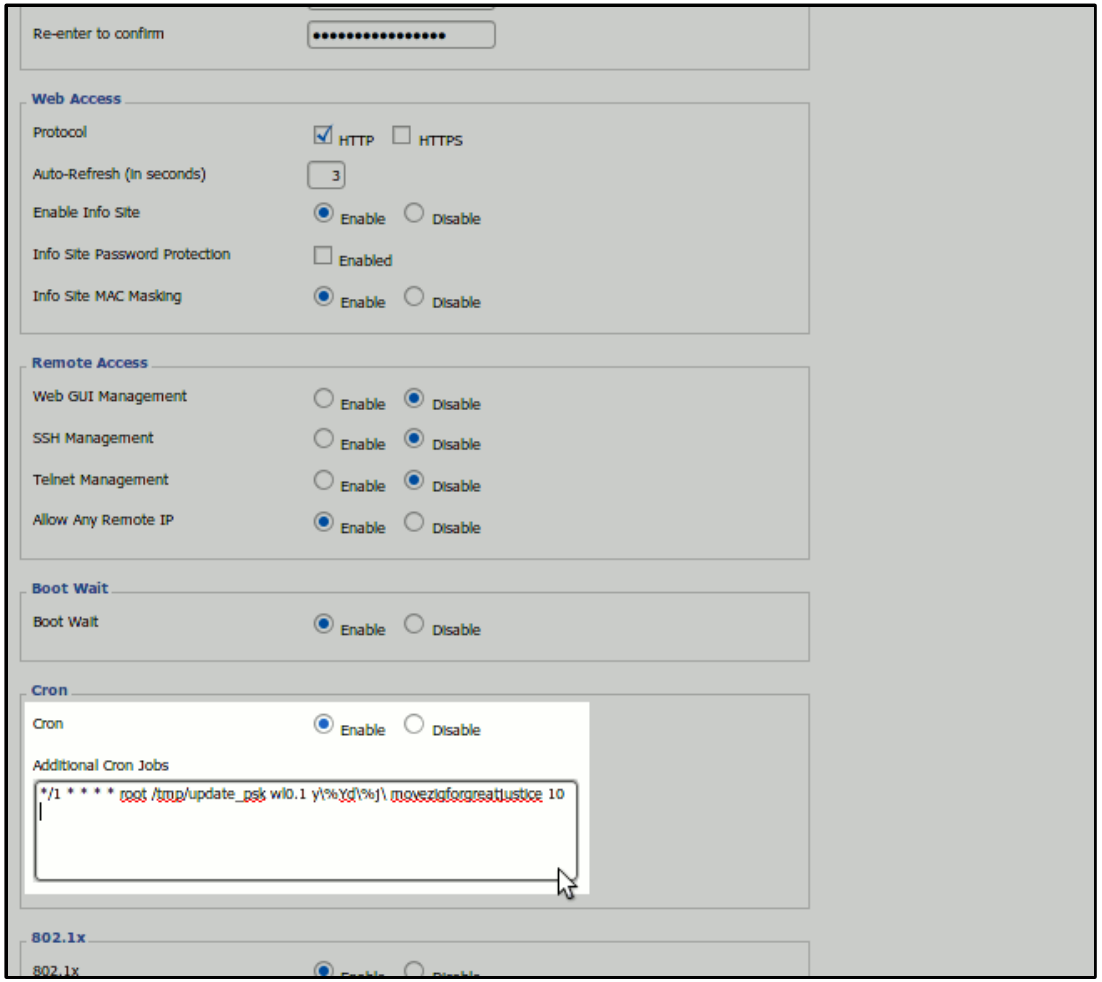
Ruhr.pm

Setup

- finally create a cronjob that executes the script
  - **Attention**: Cronjobs do not allow plain percent signs (%), as they have a special meaning. You need to escape them (replace % by \%). **Do not** use quotes around the PSK template. If you have spaces in your PSK template, escape them, too (replace “ “ by “\ “).
  - **More Attention**: DD-WRT has a bug. The cronjob is being saved correctly to NVRAM, but it displays with single quotes (') instead of backslashes (\). You must revert this before saving/applying the *Management* page again.

Ruhr.pm

Setup



Ruhr.pm

Create a Date and PSK List

```
#!/usr/bin/perl

use strict;
use warnings;

use Digest::MD5 qw(md5_hex);
use POSIX;

my $keymask
  = "y%Yd%j movezigforgreatjustice\n";
my $keylen = 10;

my $time = time;
print("Date\tKey\n");

for(1..2000){
  my $key = substr(md5_hex(strftime(
    $keymask, gmtime($time))), 0, $keylen);
  $key =~ y/0123456789/ghijkmnopq/;

  printf("%s\t%s\n", strftime('%d.%m.%Y',
    gmtime($time)), $key);

  $time += 86400;
}
```

This simple script calculates the PSKs for the next 2000 days and prints a tabstop-separated list to be read by i.e. MS Excel or OOo Calc.

You simply have to enter your key template **`$keymask`** and PSK length **`$length`**.

If you are using a template for an other validity duration than 1 day, you may want to change the **`$time`** increment and **`strftime()`** format in **`printf()`**.

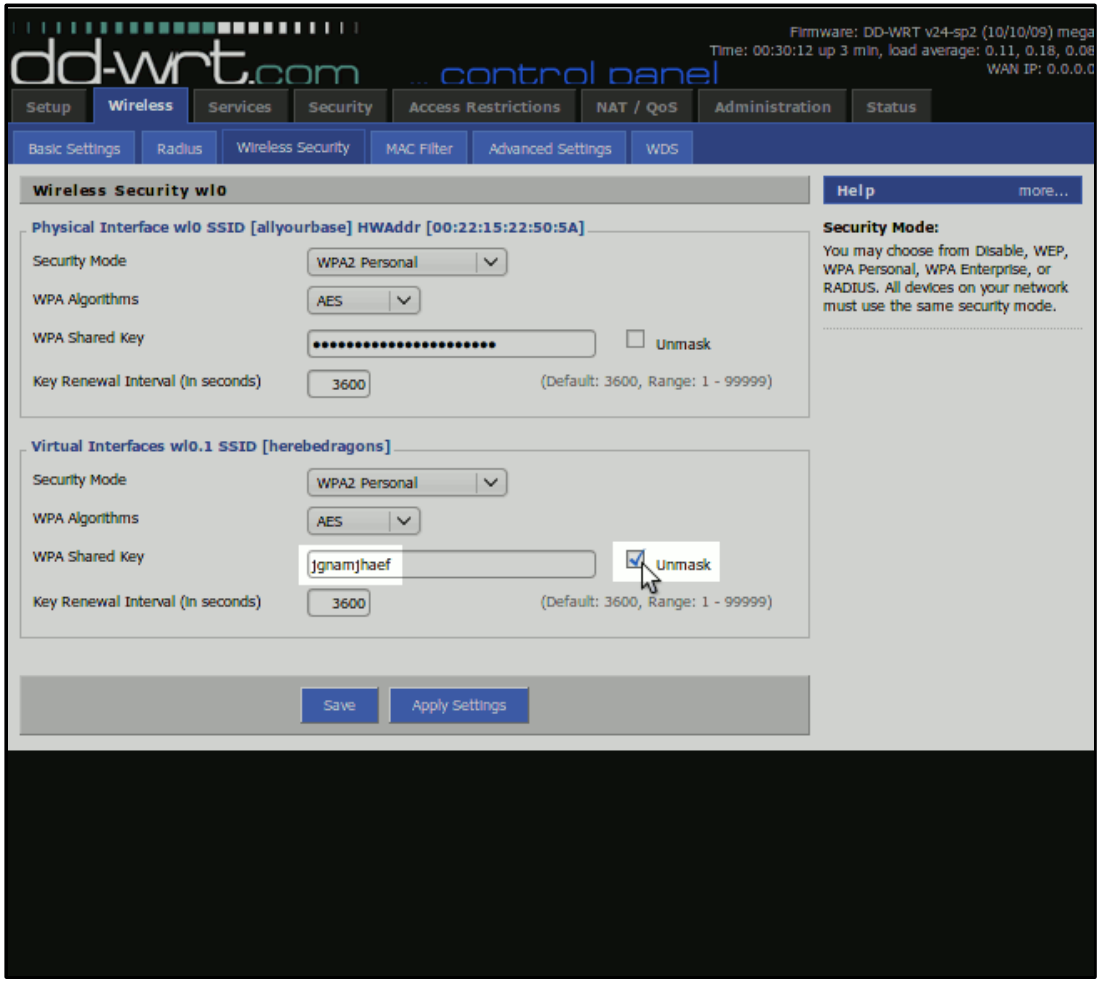
Ruhr.pm

Verify Success

- verify whether your visitors' WLAN works
  - wait a minute until the cronjob has been executed
  - navigate to the *Wireless* tab and enter the *Wireless Security* section
  - check the “unmask” checkbox of the new *Virtual Interface* device to see whether your PSK placeholder has been updated to a calculated PSK of the desired length
  - run the PSK list creation script and compare the key calculated for today with the one set on the router device

Ruhr.pm

Verify Success



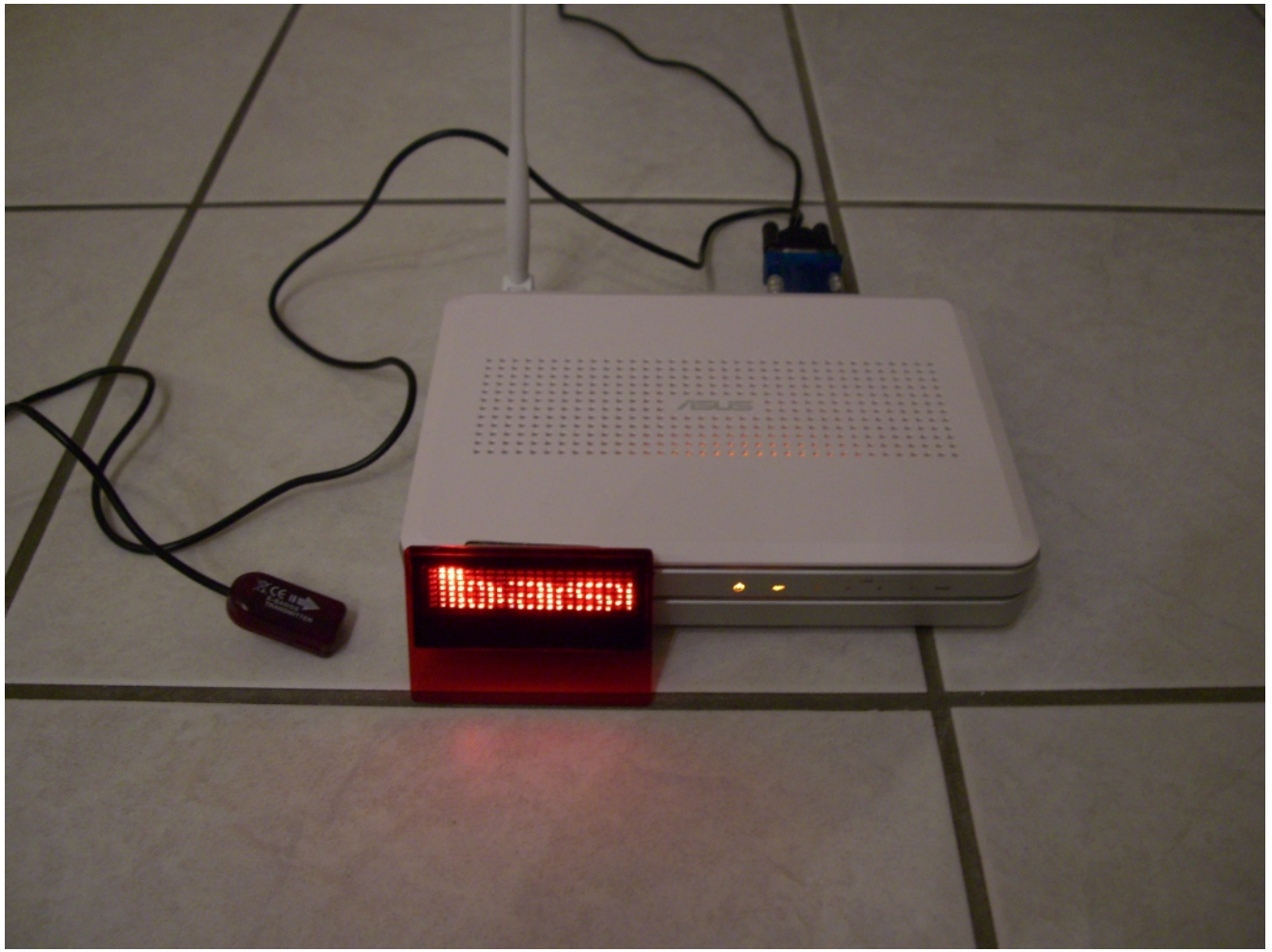
Ruhr.pm

Advanced Setups

- the following setup exists and runs in production:
  - a USB-enabled router device is equipped with a USB serial adaptor and a IR transmitter or serial cable
  - a daemon sends the current PSK every 10 seconds through the transmitter or cable
  - a LED name badge is automatically programmed with the current PSK when in range of the IR transmitter or attached using a serial cable
  - works also with electronic price labels, usually equipped with an alphanumeric LCD display

Ruhr.pm

Advanced Setups



Ruhr.pm

Advanced Setups

- a setup currently in development:
  - a USB-enabled router device is equipped with a USB-MiniUSB cable
  - a daemon detects new USB storage devices, deletes all BMP files, creates a BMP image with WLAN SSID, security configuration and PSK and stores it on the USB storage device
  - the USB storage devices are miniature electronic photo frames (2.4”) with rechargable battery, which are attached to the router to be updated, then disconnected and given away to the visitor

Ruhr.pm

Advanced Setups

Ruhr.pm

Thank you
for your attention

Ruhr.pm

Links

- DD-WRT
  - http://www.dd-wrt.com/