

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis	1
2	Voraussetzungen.....	2
2.1	Fritz!Apache	2
2.1.1	Konfiguration des Webservers	2
2.1.1.1	FritzApache.conf.....	2
2.1.1.2	Zertifikate erzeugen.....	2
2.1.2	Notwendige Konfiguration der Fritz!Box.....	2
3	Apache selbst compilieren	4
3.1	config.status	4
3.2	apache-1.3.41-modssl-mipsel-compile.sh	4
3.3	openssl-0.9.8l-mipsel-compile.sh.....	18

2 Voraussetzungen

Dynamisches DNS auf der Fritz!Box einrichten damit die Box von aussen per DNS-Namen erreichbar ist. Hierzu bei einem entsprechenden Provider einen Account einrichten:

<http://www.dyndns.com/>

<http://www.dyndnsfree.de/>

<http://dyndns.servermaster.de/>

2.1 *Fritz!Apache*

2.1.1 Konfiguration des Webservers

2.1.1.1 *FritzApache.conf*

Falls die Ports in der Konfiguration angepasst werden, dann auch die neuen Ports in den Forwardrules der ar7.cfg berücksichtigen.

- Listen 85 (http)
- Listen 543 (https)
- Server-Status aktivieren
- Server-Info aktivieren (mod_info.c muss eincompiliert werden)

2.1.1.2 Zertifikate erzeugen

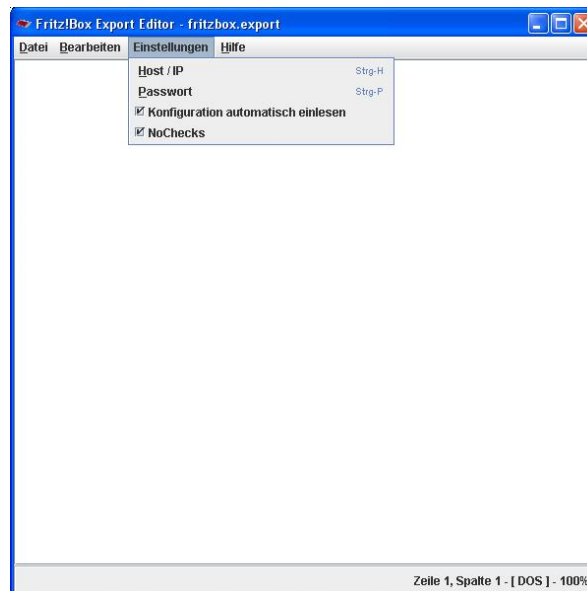


ssl-zertifikate.pdf

<http://www.schirmacher.de/display/INFO/Apache+SSL+Zertifikat+erstellen+und+installieren>

2.1.2 Notwendige Konfiguration der Fritz!Box

- DynDNS einrichten
- Mit dem FBEditor die forwardrules in der ar7.cfg anpassen, damit der Zugriff von Aussen über dynDNS möglich ist.
 - Wichtig: NoChecks markieren, sonst klappt die Rücksicherung nicht !
 - Wichtig: die gleichen Ports verwenden, sonst unterscheidet sich der Zugriff innerhalb des Netzwerkes von dem Zugriff von Aussen !



```
forwardrules = "tcp 0.0.0.0:444 0.0.0.0:444 0",  
               "udp 0.0.0.0:5060 0.0.0.0:5060",  
               "tcp 0.0.0.0:85 0.0.0.0:85 0 # Fritz!Tools an Fritz!Box (per http)",  
               "tcp 0.0.0.0:543 0.0.0.0:543 0 # Fritz!Tools an Fritz!Box (per https)";
```

3 Apache selbst compilieren

Da einige Funktionen in meiner Fritz!Box-Erweiterung nur mit Rootrechten ausgeführt werden können, muss der Apache mit einigen Modifikationen selbst compiliert werden. Hier das Vorgehen:

- Freetz installieren und eine eigene Minimal-Version bauen
- Das Freetz-Verzeichnis inkl. der Toolchain in ein eigenes Verzeichnis kopieren oder später die Freetz-Pfade nutzen
- ein Apache-Verzeichnis anlegen
- Folgende Dateien hatte ich heruntergeladen und anpasst <http://www.ip-phone-forum.de/attachment.php?attachmentid=15573&d=1178410545>
 - config.status
 - openssl-0.9.81-mipsel-compile.sh
 - apache-1.3.41-modssl-mipsel-compile.sh

Das ganze habe ich auf einem Rechner mit Ubuntu durchgeführt.

3.1 config.status

```
#!/bin/sh
##
## config.status -- APACI auto-generated configuration restore script
##
## Use this shell script to re-run the APACI configure script for
## restoring your configuration. Additional parameters can be supplied.
##

CC="/home/peter/fritzbox/apache/./devenv/toolchain/target/bin/mipsel-linux-uclibc-gcc" \
CFLAGS="-Os -W -Wall -pipe -march=mips32 -mips32 -Wa,--trap -msoft-float -DBIG_SECURITY_HOLE" \
LDFLAGS="-static-libgcc" \
TARGET="apache" \
SSL_BASE="/home/peter/fritzbox/apache/openssl-0.9.81" \
./configure \
"--with-layout=Apache" \
"--target=apache" \
"--prefix=apache-1.3.41-modssl" \
"--enable-module=info" \
"--enable-module=rewrite" \
"--enable-module=vhost_alias" \
"--enable-module=speling" \
"--enable-module=ssl" \
"$@"
```

3.2 apache-1.3.41-modssl-mipsel-compile.sh

```
#!/bin/bash

# see: http://www.modssl.org/example/

# it works!

HOME=/home/peter/fritzbox/apache
NAME=apache-1.3.41-modssl

SRC=apache_1.3.41.tar.gz

URL=http://apache.mirror.iphh.net/httpd/$SRC

SSL_SRC=mod_ssl-2.8.31-1.3.41.tar.gz
```

```

SSL_URL=http://www.modssl.org/source/$SSL_SRC
PATCHFILE=apache-1.3.41-mipsel.patch
CC_PATH=$HOME/..devenv/toolchain/target/bin
SRC_PATH=$HOME/apache_1.3.41
SRC_PATH_SSL=$HOME/mod_ssl-2.8.31-1.3.41
BIN_PATH_RELATIVE=$NAME
BIN_PATH=$SRC_PATH/$BIN_PATH_RELATIVE
OUTPUT_BZIP=$SRC_PATH/./$NAME-mipsel.tar.bz2
OPENSSL_PATH=$HOME/openssl-0.9.8l
OPENSSL_BINARY_ARCHIVE=openssl-0.9.8l-mipsel.tar.bz2
OPENSSL_COMPILE_SCRIPT=openssl-0.9.8l-mipsel-compile.sh

cd $HOME

echo "Checking for dependencies..."
# check for openssl
if test ! -d $OPENSSL_PATH; then
    echo "Need to get openssl first. Checking for mipsel binary archive $OPENSSL_BINARY_ARCHIVE";
    if test -f $OPENSSL_BINARY_ARCHIVE; then
        echo "Found $OPENSSL_BINARY_ARCHIVE, unpacking...";
        unq $OPENSSL_BINARY_ARCHIVE
        if test ! -d $OPENSSL_PATH; then
            echo "Could not find openssl path. Exiting.";
            exit 1;
        fi
    else
        echo "Did not find $OPENSSL_BINARY_ARCHIVE. Need to compile openssl first.";
        if test -x $OPENSSL_COMPILE_SCRIPT; then
            if ./$OPENSSL_COMPILE_SCRIPT; then
                if test -f $OPENSSL_BINARY_ARCHIVE; then
                    echo "Found $OPENSSL_BINARY_ARCHIVE, unpacking...";
                    unq $OPENSSL_BINARY_ARCHIVE
                    if test ! -d $OPENSSL_PATH; then
                        echo "Could not find openssl path. Exiting.";
                        exit 1;
                    fi
                else
                    echo "Could not find openssl binary archive. Exiting.";
                fi
            fi
        fi
    fi

```

```

        exit 1;
    fi
else
    echo "openssl make failed.";
    exit 1;
fi
else
    echo "Could not find openssl compile script. Need $OPENSSL_COMPILE_SCRIPT.";
    exit 1;
fi
fi
fi
echo "Finished checking for dependencies.";

# the following is for apache
if test ! -f $SRC; then wget $URL; fi
if test ! -f $SRC; then echo "Could not find file $SRC"; exit 1; fi

if test -d $SRC_PATH; then rm -rf $SRC_PATH; fi
unp $SRC

if test ! -d $SRC_PATH; then echo "Could not find source path: $SRC_PATH"; exit 1; fi
cd $SRC_PATH

cat > $PATCHFILE <<\ENDPATCH
--- src/include/ap_config.h.old    2007-02-24 15:50:55.000000000 +0100
+++ src/include/ap_config.h        2007-02-24 15:50:45.000000000 +0100
@@ -1459,4 +1459,6 @@
}
#endif

+#undef SYS_SIGLIST
+
#endif /* !AP_CONFIG_H */
--- /dev/null    2006-10-26 17:26:42.000000000 +0200
+++ src/main/uri_delims.h    2007-02-24 16:05:49.000000000 +0100
@@ -0,0 +1,17 @@
+/* this file is automatically generated by gen_uri_delims, do not edit */

```



```

+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*0x50...0x57*/
+ 0x00, 0x00, 0x00, 0x0f, 0x1f, 0x0f, 0x07, 0x00, /*0x58...0x5f*/
+ 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*0x60...0x67*/
+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*0x68...0x6f*/
+ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /*0x70...0x77*/
+ 0x00, 0x00, 0x00, 0x0f, 0x27, 0x0f, 0x01, 0x3e, /*0x78...0x7f*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0x80...0x87*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0x88...0x8f*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0x90...0x97*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0x98...0x9f*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xa0...0xa7*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xa8...0xaf*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xb0...0xb7*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xb8...0xbf*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xc0...0xc7*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xc8...0xcf*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xd0...0xd7*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xd8...0xdf*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xe0...0xe7*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xe8...0xef*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, /*0xf0...0xf7*/
+ 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36, 0x36 /*0xf8...0xff*/
+
+};
+
--- src/Configure.old      2006-07-12 09:16:06.000000000 +0200
+++ src/Configure         2007-02-24 16:08:49.000000000 +0100
@@ -1582,13 +1582,16 @@
case "$PLAT" in
  *-linux*)
    # newer systems using glibc 2.x need -lcrypt
-   if ./helpers/TestCompile lib crypt; then
+   #if ./helpers/TestCompile lib crypt; then
+   if true; then
      LIBS="$LIBS -lcrypt"
    fi
    # see if prctl(PR_SET_DUMPABLE) is available
-   if TCADDINCL=#include <sys/prctl.h>

```



```

-static int required_flag = PR_SET_DUMPABLE; ./helpers/TestCompile sizeof required_flag ; then
-     if ./helpers/TestCompile func prctl; then
+#     if TCADDINCL=#include <sys/prctl.h>
+#static int required_flag = PR_SET_DUMPABLE; ./helpers/TestCompile sizeof required_flag ; then
+     if true; then
+         #if ./helpers/TestCompile func prctl; then
+             if true; then
+                 CFLAGS="$CFLAGS -DHAVE_SET_DUMPABLE"
+             fi
+         fi
@@ -1597,12 +1600,14 @@
*_dg-dgux*)
    # R4.11MU02 requires -lsocket -lnsl ... no idea if it's earlier or
    # later than what we already knew about. PR#732
-    if ./helpers/TestCompile lib nsl; then
+    #if ./helpers/TestCompile lib nsl; then
+    if true; then
+        LIBS="$LIBS -lnsl"
+        TLIB='-lnsl'
+    fi
-    if TLIB=$TLIB ./helpers/TestCompile lib socket; then
-        LIBS="-lsocket $LIBS"
+    #if TLIB=$TLIB ./helpers/TestCompile lib socket; then
+    if false; then
+        LIBS="-lsocket $LIBS"
+    fi
    ;;
BS2000*-siemens-sysv4*)
@@ -1619,34 +1624,41 @@
    # -lnsl may be available (or may be not).
    # In standard SVR4 systems, -lsocket relies on some symbols
    # from -lnsl, so the test for -lnsl must appear first.
-    if ./helpers/TestCompile lib nsl; then
+    #if ./helpers/TestCompile lib nsl; then
+    if true; then
+        LIBS="$LIBS -lnsl"
+        TLIB='-lnsl'
+    fi

```

```

-   if TLIB=$TLIB ./helpers/TestCompile lib socket; then
+   #if TLIB=$TLIB ./helpers/TestCompile lib socket; then
+   if false; then
+       LIBS="-lsocket $LIBS"
+   fi
+   # Auto-detect presence of libdl for dynamic loading
-   if ./helpers/TestCompile lib dl; then
-       if ./helpers/TestCompile func dlopen; then
-           LIBS="$LIBS -ldl"
-           TLIB='-ldl'
+   #if ./helpers/TestCompile lib dl; then
+   if true; then
+       #if ./helpers/TestCompile func dlopen; then
+           if false; then
+               LIBS="$LIBS -ldl"
+               TLIB='-ldl'
+           fi
+       fi
+       # Test for the presence of the "union semun":
-       if TCADDINCL=#include <sys/types.h>
+#       if TCADDINCL=#include <sys/types.h>
+       #include <sys/ipc.h>
+       #include <sys/sem.h>' ./helpers/TestCompile sizeof "union semun"; then
+       if false; then
+           : Okay, union semun is defined
+       else
+           CFLAGS="$CFLAGS -DNEED_UNION_SEMUN"
+       fi
+       # Test for the presence of the _rini_struct typedef:
-       if TCADDINCL=#include <pwd.h>' ./helpers/TestCompile sizeof _rini_struct; then
+       #if TCADDINCL=#include <pwd.h>' ./helpers/TestCompile sizeof _rini_struct; then
+       if false; then
+           CFLAGS="$CFLAGS -DHAVE_RINI_STRUCTURE"
+       fi
+       # Test whether initgroups() must be emulated:
-       if ./helpers/TestCompile func initgroups; then
+       #if ./helpers/TestCompile func initgroups; then
+       if true; then

```

```

:
else
    CFLAGS="$CFLAGS -DNEED_INITGROUPS"
@@ -1882,7 +1894,8 @@
fi

if [ "$SRULE_EXPAT" = "xyes" ]; then
- if ./helpers/TestCompile lib expat; then
+ #if ./helpers/TestCompile lib expat; then
+     if false; then
        echo " + using system Expat"
        LIBS="$LIBS -lexpat"
    else
@@ -2186,19 +2199,23 @@
        esac
        ;;
        *-hp-hpux*)
-         if ./helpers/TestCompile func shl_load; then
+         #if ./helpers/TestCompile func shl_load; then
+             if false; then
                :
            else
-             if ./helpers/TestCompile lib dld; then
+             #if ./helpers/TestCompile lib dld; then
+             if false; then
                DL_LIB="-ldld"
            fi
        fi
        ;;
        *)
-         if ./helpers/TestCompile func dlopen; then
+         #if ./helpers/TestCompile func dlopen; then
+             if false; then
                :
            else
-             if ./helpers/TestCompile lib dl; then
+             #if ./helpers/TestCompile lib dl; then
+             if false; then

```

```

DL_LIB="-ldl"

fi

fi

@@ -2216,7 +2233,8 @@

## We check the sizeof various data types

##

echo " + checking sizeof various data types"

-AP_TYPE_QUAD=`./helpers/TestCompile -r sizeof 'long long'^
+#AP_TYPE_QUAD=`./helpers/TestCompile -r sizeof 'long long'^
+AP_TYPE_QUAD=8

if [ "x$AP_TYPE_QUAD" = "x" ]; then
    AP_TYPE_QUAD="unknown_quad"
    AP_LONGEST_LONG="long"
@@ -2235,13 +2253,15 @@

##

## Check for availability of isinf() and isnan()

##

-if ./helpers/TestCompile func isinf ; then
+#if ./helpers/TestCompile func isinf ; then
+if false; then
    echo "" >>$AP_CONFIG_AUTO_H

    echo "/* determine: isinf() found in libc */ " >>$AP_CONFIG_AUTO_H

    echo "#ifndef HAVE_ISINF" >>$AP_CONFIG_AUTO_H

    echo "#define HAVE_ISINF 1" >>$AP_CONFIG_AUTO_H

    echo "#endif" >>$AP_CONFIG_AUTO_H

-elif ./helpers/TestCompile lib m isinf ; then
+#elif ./helpers/TestCompile lib m isinf ; then
+elif false; then
    echo "" >>$AP_CONFIG_AUTO_H

    echo "/* determine: isinf() found in libm */ " >>$AP_CONFIG_AUTO_H

    echo "#ifndef HAVE_ISINF" >>$AP_CONFIG_AUTO_H

@@ -2251,13 +2271,15 @@

    ADDED_LM="yes"

fi

-if ./helpers/TestCompile func isnan ; then
+#if ./helpers/TestCompile func isnan ; then
+if false; then

```

```

echo "" >>$AP_CONFIG_AUTO_H

echo "/* determine: isnan() found in libc */" >>$AP_CONFIG_AUTO_H

echo "#ifndef HAVE_ISNAN" >>$AP_CONFIG_AUTO_H

echo "#define HAVE_ISNAN 1" >>$AP_CONFIG_AUTO_H

echo "#endif" >>$AP_CONFIG_AUTO_H

-elif ./helpers/TestCompile lib m isnan ; then
+elif ./helpers/TestCompile lib m isnan ; then

+elif false; then

echo "" >>$AP_CONFIG_AUTO_H

echo "/* determine: isnan() found in libm */" >>$AP_CONFIG_AUTO_H

echo "#ifndef HAVE_ISNAN" >>$AP_CONFIG_AUTO_H

@@ -2270,7 +2292,8 @@

## We check for the endianness of the machine

##

-AP_BYTE_ORDER=`./helpers/TestCompile -r byteorder`
+AP_BYTE_ORDER=`./helpers/TestCompile -r byteorder`
+AP_BYTE_ORDER=12

if [ "x$AP_BYTE_ORDER" = "x21" ]; then
    AP_BYTE_ORDER="21" # big endian
else
@@ -2286,12 +2309,14 @@

##

## Now compare the sizes of off_t to long

##

-AP_TYPE_OFF_T=`./helpers/TestCompile -r sizeof off_t`
+AP_TYPE_OFF_T=`./helpers/TestCompile -r sizeof off_t`
+AP_TYPE_OFF_T=4

if [ "x$AP_TYPE_OFF_T" = "x" ]; then
    AP_TYPE_OFF_T="unknown_off_t"
fi

-AP_TYPE_LONG=`./helpers/TestCompile -r sizeof long`
+AP_TYPE_LONG=`./helpers/TestCompile -r sizeof long`
+AP_TYPE_LONG=4

if [ "x$AP_TYPE_LONG" = "x" ]; then
    AP_TYPE_LONG="unknown_long"
fi

```

```

@@ -2313,7 +2338,8 @@

##

## Now see of void * is as big as a quad (long long)

##

-AP_TYPE_VOID_P=`./helpers/TestCompile -r sizeof 'void *'^
+AP_TYPE_VOID_P=`./helpers/TestCompile -r sizeof 'void *'^
+AP_TYPE_VOID_P=4
if [ "x$AP_TYPE_VOID_P" = "x" ]; then
    AP_TYPE_VOID_P="unknown_void_p"
fi

--- src/main/Makefile.tmpl.old    2004-11-24 20:10:20.000000000 +0100
+++ src/main/Makefile.tmpl      2007-02-24 16:12:11.000000000 +0100

@@ -23,20 +23,23 @@

    ar cr $@ $(OBJS)

    $(RANLIB) $@

+clean:

+#   rm -f *.o $(LIB) uri_delims.h gen_uri_delims test_char.h gen_test_char
+
clean:
-   rm -f *.o $(LIB) uri_delims.h gen_uri_delims test_char.h gen_test_char
+   rm -f *.o $(LIB) gen_uri_delims gen_test_char

distclean: clean

    -rm -f Makefile

-uri_delims.h: gen_uri_delims
-   ./gen_uri_delims >uri_delims.h
+#uri_delims.h: gen_uri_delims
+#   ./gen_uri_delims >uri_delims.h

gen_uri_delims: gen_uri_delims.o

    $(CC) $(CFLAGS) $(LDFLAGS) -o gen_uri_delims gen_uri_delims.o $(LIBS)

-test_char.h: gen_test_char
-   ./gen_test_char >test_char.h
+#test_char.h: gen_test_char
+#   ./gen_test_char >test_char.h

```

```

gen_test_char: gen_test_char.o
        $(CC) $(CFLAGS) $(LDFLAGS) -o gen_test_char gen_test_char.o $(LIBS)

ENDPATCH

if \
patch -N -p0 -i $PATCHFILE \
; then ;; else echo "Could not apply patch: $PATCHFILE"; exit 1; fi
rm -f $PATCHFILE

cd $HOME

# then we prepare mod_ssl
if test ! -f $SSL_SRC; then wget $SSL_URL; fi
if test ! -f $SSL_SRC; then      echo "Could not find file $SSL_SRC"; exit 1; fi

if test -d $SRC_PATH_SSL; then rm -rf $SRC_PATH_SSL; fi
unp $SSL_SRC

if test ! -d $SRC_PATH_SSL; then echo "Could not find source path: $SRC_PATH_SSL"; exit 1; fi
cd $SRC_PATH_SSL

# problem: need special patch tool that is cross compiled... so we need to make sure it is compiled for 386
cd etc;
unp patch.tar;
cd patch;
CC="/usr/bin/gcc" ./configure
make patch
cd ../..;

#the first configure fails because the patch program is cross compiled as well
if \
CC="$CC_PATH/mipsel-linux-uclibc-gcc" \
LD="$CC_PATH/mipsel-linux-uclibc-ld" \
CFLAGS="-Os -W -Wall -pipe -march=mips32 -mips32 -Wa,--trap -msoft-float -DBIG_SECURITY_HOLE" \
LDFLAGS="-static-libgcc" \
./configure \

```

```

--with-layout=Apache \
--with-patch=etc/patch/patch \
--with-apache=$SRC_PATH \
--with-ssl=$OPENSSL_PATH \
--target=apache \
--prefix=$BIN_PATH_RELATIVE \
--enable-module=info \
--enable-module=rewrite \
--enable-module=vhost_alias \
--enable-module=speling \
; then ;; else echo "configure for mod_ssl failed."; exit 1; fi

# switching back to apache
cd $SRC_PATH

if \
make \
; then ;; else echo "make failed."; exit 1; fi

# we need to patch make certificate since the target openssl binary cannot be executed.
# instead we use the local machine openssl this is interactive!

cat > $PATCHFILE <<\ENDPATCH
--- src/Makefile.old 2007-05-04 12:23:42.000000000 +0000
+++ src/Makefile 2007-05-04 12:22:27.000000000 +0000
@@ -40,7 +40,7 @@
SSL_BINDIR=$(SSL_BASE)/bin
SSL_INCDIR=$(SSL_BASE)/include
SSL_LIBDIR=$(SSL_BASE)/lib
-SSL_PROGRAM=/home/bofh/openssl-0.9.8e/bin/openssl
+SSL_PROGRAM=/usr/bin/openssl
SSL_VERSION=-DMOD_SSL_VERSION="2.8.31"
SSL_CFLAGS= -DSSL_COMPAT -DSSL_USE_SDBM -I$(SSL_INCDIR)
SSL_VENDOR_OBJS=
ENDPATCH

# if \

```



```

# patch -N -p0 -i $PATCHFILE \
# ; then ;; else echo "Could not apply patch: $PATCHFILE"; exit 1; fi
# rm -f $PATCHFILE

# if \
# make certificate \
# ; then ;; else echo "make certificate failed."; exit 1; fi

# we do not need openssl if we want to create a dummy certificate
if \
make certificate TYPE=dummy \
; then ;; else echo "make certificate failed."; exit 1; fi

if \
make install \
; then ;; else echo "make install failed."; exit 1; fi

rm -rf $BIN_PATH/man
rm -rf $BIN_PATH/htdocs/*
rm -rf $BIN_PATH/cgi-bin/*
rm -rf $BIN_PATH/conf/*default
rm -rf $BIN_PATH/include
rm -rf $BIN_PATH/icons
rm -rf $BIN_PATH/libexec
rm -rf $BIN_PATH/proxy

for f in $BIN_PATH/bin/* ; do
    if file $f | grep -q "ELF"; then
        FILESIZE1=$(stat -c%s "$f");
        if \
        $CC_PATH/mipsel-linux-uclibc-strip --remove-section=.comment --remove-section=.note $f \
        ; then ;; else echo "strip for $f failed."; fi
        FILESIZE2=$(stat -c%s "$f");
        echo "$FILESIZE1 $f";
        echo $FILESIZE2;
    fi
done

```

```
tar -cjf $OUTPUT_BZIP $BIN_PATH_RELATIVE
if test ! -f $OUTPUT_BZIP; then echo "Could not find file $OUTPUT_BZIP"; exit 1; fi
```

```
cd $HOME
```

```
#rm -f $SRC;
#rm -f $SSL_SRC;
#rm -rf $SRC_PATH_SSL;
#rm -rf $OPENSSL_PATH;
#rm -rf $SRC_PATH;
```

3.3 openssl-0.9.8l-mipsel-compile.sh

```
#!/bin/bash
```

```
HOME=/home/peter/fritzbox/apache
```

```
NAME=openssl-0.9.8l
```

```
SRC=openssl-0.9.8l.tar.gz
```

```
URL=http://www.openssl.org/source/$SRC
```

```
PATCHFILE=openssl-0.9.8l-mipsel.patch
```

```
CC_PATH=$HOME/../../devenv/toolchain/target/bin
```

```
SRC_PATH=$HOME/openssl-0.9.8l
```

```
BIN_PATH_RELATIVE=$NAME
```

```
BIN_PATH=$SRC_PATH/$BIN_PATH_RELATIVE
```

```
OUTPUT_BZIP=$HOME/$NAME-mipsel.tar.bz2
```

```
cd $HOME
```

```
if test ! -f $SRC; then wget $URL; fi
```

```
if test ! -f $SRC; then echo "Could not find file $SRC"; exit 1; fi
```

```
if test -d $SRC_PATH; then rm -rf $SRC_PATH; fi
```

```
unp $SRC
```

```
if test ! -d $SRC_PATH; then echo "Could not find source path: $SRC_PATH"; exit 1; fi
```

```
cd $SRC_PATH
```

```
cat > $PATCHFILE <<ENDPATCH
```

```
--- Configure.old 2007-05-03 20:56:32.000000000 +0000
```

```
+++ Configure 2007-05-03 21:01:43.000000000 +0000
```

```
@ -138,6 +138,7 @@
```

```
# File 'TABLE' (created by 'make TABLE') contains the data from this list,
```

```
# formatted for better readability.
```

```
+ "mipsel-linux", "mipsel-linux-uclibc-gcc:-Os -W -Wall -pipe -march=mips32 -mips32 -Wa,--trap -msoft-float:::pthread:::so:",
```

```
#"b", "$ {tcc};$ {tflags}::$ {tlib};$ {bits1};$ {tbn_mul}:::",
```

```
#"bl-4c-2c", "$ {tcc};$ {tflags}::$ {tlib};$ {bits1}BN_LLONG RC4_CHAR MD2_CHAR:$ {tbn_mul}:::",
```

```
ENDPATCH
```

```
if \
```

```
patch -N -p0 -i $PATCHFILE \
```

```
; then ;; else echo "Could not apply patch: $PATCHFILE"; exit 1; fi
```

```
rm -f $PATCHFILE
```

```
if \
```

```
CC="$CC_PATH/mipsel-linux-uclibc-gcc" \
```

```
LD="$CC_PATH/mipsel-linux-uclibc-ld" \
```

```
CFLAGS="-Os -W -Wall -pipe -march=mips32 -mips32 -Wa,--trap -msoft-float" \
```

```
LDLFLAGS="-static-libgcc" \
```

```
./Configure \
```

```
--openssldir=$BIN_PATH \
```

```
mipsel-linux \
```

```
; then ;; else echo "configure failed."; exit 1; fi
```

```
if \
```

```
make \
```

```
; then ;; else echo "make failed."; exit 1; fi
```

```
if \
```

```
make install \
```

```
; then ;; else echo "make install failed."; exit 1; fi
```

```
rm -rf $BIN_PATH/man
```

```
rm -rf $BIN_PATH/private
```

```
rm -rf $BIN_PATH/certs
```

```
for f in $BIN_PATH/bin/* $BIN_PATH/lib/* ; do
    if file $f | grep -q "ELF"; then
        FILESIZE1=$(stat -c%s "$f");
        if \
            $CC_PATH/mipsel-linux-uclibc-strip --remove-section=.comment --remove-section=.note $f \
            ; then ;; else echo "strip for $f failed."; fi
        FILESIZE2=$(stat -c%s "$f");
        echo "$FILESIZE1 $f";
        echo $FILESIZE2;
    fi
done

tar -cjf $OUTPUT_BZIP $BIN_PATH_RELATIVE

if test ! -f $OUTPUT_BZIP; then echo "Could not find file $OUTPUT_BZIP"; exit 1; fi

cd $HOME

rm -f $SRC
# rm -rf $SRC_PATH
```