****
1265 Fri May 16 10:29:50 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/p12split/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
****

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #

  25 include ../Makefile.com

  27 PROG=           p12split
  28 LDLIBS +=       -lwanboot -linetutil -lwanbootutil
  28 LDLIBS +=       -lwanboot -linetutil -lwanbootutil -lcrypto
  29 CPPFLAGS +=     -I$(CMNCRYPTDIR)

  31 # libcrypto has no lint library, so we can only include this while building
  32 $(PROG) := LDLIBS += -lcrypto

  34 LINTFLAGS +=    -erroff=E_NAME_USED_NOT_DEF2
  35 #endif /* ! codereview */

  37 all:            $(PROG)

  39 install:        all $(ROOTCMD)

  41 clean:

  43 lint:           lint_PROG

  45 include ../../../../Makefile.targ
```

```
**********************************************************
   15474 Fri May 16 10:29:50 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/p12split/p12split.c
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
/*
 * CDDL HEADER START
 *
 * The contents of this file are subject to the terms of the
 * Common Development and Distribution License, Version 1.0 only
 * (the "License").  You may not use this file except in compliance
 * with the License.
 *
 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
 * or http://www.opensolaris.org/os/licensing.
 * See the License for the specific language governing permissions
 * and limitations under the License.
 *
 * When distributing Covered Code, include this CDDL HEADER in each
 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 * If applicable, add the following below this CDDL HEADER, with the
 * fields enclosed by brackets "[]" replaced with your own identifying
 * information: Portions Copyright [yyyy] [name of copyright owner]
 *
 * CDDL HEADER END
 */
/*
 * Copyright 2002-2003 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms.
 */

#pragma ident   "%Z%%M% %I%     %E% SMI"

#include <stdio.h>
#include <libintl.h>
#include <locale.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/wanboot_impl.h>
#include <unistd.h>
#include <string.h>
#include <libinetutil.h>
#include <wanbootutil.h>

#include <openssl/crypto.h>
#include <openssl/buffer.h>
#include <openssl/bio.h>
#include <openssl/err.h>
#include <openssl/x509.h>
#include <openssl/x509v3.h>
#include <openssl/pkcs12.h>
#include <openssl/evp.h>
#include <p12aux.h>

static boolean_t verbose = B_FALSE;     /* When nonzero, do in verbose mode */

/* The following match/cert values require PKCS12 */
static int  matchty;            /* Type of matching do to on input */
static char *k_matchval;        /* localkeyid value to match */
static uint_t k_len;            /* length of k_matchval */

#define IO_KEYFILE      1       /* Have a separate key file or data */
#define IO_CERTFILE     2       /* Have a separate cert file or data */
#define IO_TRUSTFILE    4       /* Have a separate trustanchor file */
```



```
static char *input = NULL;      /* Consolidated input file */
static char *key_out = NULL;    /* Key file to be output */
static char *cert_out = NULL;   /* Cert file to be output */
static char *trust_out = NULL;  /* Trust anchor file to be output */
static uint_t outfiles;         /* What files are there for output */
static char *progname;

/* Returns from time_check */
typedef enum {
        CHK_TIME_OK = 0,                /* Cert in effect and not expired */
        CHK_TIME_BEFORE_BAD,            /* not_before field is invalid */
        CHK_TIME_AFTER_BAD,             /* not_after field is invalid */
        CHK_TIME_IS_BEFORE,             /* Cert not yet in force */
        CHK_TIME_HAS_EXPIRED            /* Cert has expired */
} time_errs_t;
_____unchanged_portion_omitted_

static int
do_certs(void)
{
        char *bufp;
        STACK_OF(X509) *ta_in = NULL;
        EVP_PKEY *pkey_in = NULL;
        X509 *xcert_in = NULL;

        sunw_crypto_init();

        if (read_files(&ta_in, &xcert_in, &pkey_in) < 0)
                return (-1);

        if (verbose) {
                if (xcert_in != NULL) {
                        (void) printf(gettext("\nMain cert:\n"));

                        /*
                         * sunw_subject_attrs() returns a pointer to
                         * memory allocated on our behalf. The same
                         * behavior is exhibited by sunw_issuer_attrs().
                         */
                        bufp = sunw_subject_attrs(xcert_in, NULL, 0);
                        if (bufp != NULL) {
                                (void) printf(gettext("  Subject: %s\n"),
                                    bufp);
                                OPENSSL_free(bufp);
                        }

                        bufp = sunw_issuer_attrs(xcert_in, NULL, 0);
                        if (bufp != NULL) {
                                (void) printf(gettext("  Issuer: %s\n"), bufp);
                                OPENSSL_free(bufp);
                        }

                        (void) sunw_print_times(stdout, PRNT_BOTH, NULL,
                            xcert_in);
                }

                if (ta_in != NULL) {
                        X509 *x;
                        int i;

                        for (i = 0; i < sk_X509_num(ta_in); i++) {
                                /* LINTED */
                                x = sk_X509_value(ta_in, i);
                                (void) printf(
                                    gettext("\nTrust Anchor cert %d:\n"), i);
```

```
267                                 /*
268                                  * sunw_subject_attrs() returns a pointer to
269                                  * memory allocated on our behalf. We get the
270                                  * same behavior from sunw_issuer_attrs().
271                                  */
272                                 bufp = sunw_subject_attrs(x, NULL, 0);
273                                 if (bufp != NULL) {
274                                         (void) printf(
275                                             gettext("  Subject: %s\n"), bufp);
276                                         OPENSSL_free(bufp);
277                                 }

279                                 bufp = sunw_issuer_attrs(x, NULL, 0);
280                                 if (bufp != NULL) {
281                                         (void) printf(
282                                             gettext("  Issuer: %s\n"), bufp);
283                                         OPENSSL_free(bufp);
284                                 }

286                                 (void) sunw_print_times(stdout, PRNT_BOTH,
287                                     NULL, x);
288                         }
289                 }
290         }

292         check_certs(ta_in, &xcert_in);
293         if (xcert_in != NULL && pkey_in != NULL) {
294                 if (sunw_check_keys(xcert_in, pkey_in) == 0) {
295                         wbku_printerr("warning: key and certificate do "
296                             "not match\n");
297                 }
298         }

300         return (write_files(ta_in, xcert_in, pkey_in));
301 }
```

*unchanged_portion_omitted_*

```
354 static void
355 check_certs(STACK_OF(X509) *ta_in, X509 **c_in)
356 {
357         X509 *curr;
358         time_errs_t ret;
359         int i;
360         int del_expired = (outfiles != 0);

362         if (c_in != NULL && *c_in != NULL) {
363                 ret = time_check_print(*c_in);
364                 if ((ret != CHK_TIME_OK && ret != CHK_TIME_IS_BEFORE) &&
365                     del_expired) {
366                         (void) fprintf(stderr, gettext("  Removing cert\n"));
367                         X509_free(*c_in);
368                         *c_in = NULL;
369                 }
370         }

372         if (ta_in == NULL)
373                 return;

375         for (i = 0; i < sk_X509_num(ta_in); ) {
379                 /* LINTED */
376                 curr = sk_X509_value(ta_in, i);
377                 ret = time_check_print(curr);
378                 if ((ret != CHK_TIME_OK && ret != CHK_TIME_IS_BEFORE) &&
379                     del_expired) {
380                         (void) fprintf(stderr, gettext("  Removing cert\n"));
```



```
385                         /* LINTED */
381                         curr = sk_X509_delete(ta_in, i);
382                         X509_free(curr);
383                         continue;
384                 }
385                 i++;
386         }
387 }
```

*unchanged_portion_omitted_*

```
521 static int
522 do_ofile(char *name, EVP_PKEY *pkey, X509 *cert, STACK_OF(X509) *ta)
523 {
524         STACK_OF(EVP_PKEY) *klist = NULL;
525         STACK_OF(X509)  *clist = NULL;
526         PKCS12          *p12 = NULL;
527         int             ret = 0;
528         FILE            *fp;
529         struct stat     sbuf;

531         if (stat(name, &sbuf) == 0 && !S_ISREG(sbuf.st_mode)) {
532                 wbku_printerr("%s is not a regular file\n", name);
533                 return (-1);
534         }

536         if ((fp = fopen(name, "w")) == NULL) {
537                 wbku_printerr("cannot open output file %s", name);
538                 return (-1);
539         }

541         if ((clist = sk_X509_new_null()) == NULL ||
542             (klist = sk_EVP_PKEY_new_null()) == NULL) {
543                 wbku_printerr("out of memory\n");
544                 ret = -1;
545                 goto cleanup;
546         }

548         if (cert != NULL && sk_X509_push(clist, cert) == 0) {
549                 wbku_printerr("out of memory\n");
550                 ret = -1;
551                 goto cleanup;
552         }

554         if (pkey != NULL && sk_EVP_PKEY_push(klist, pkey) == 0) {
555                 wbku_printerr("out of memory\n");
556                 ret = -1;
557                 goto cleanup;
558         }

560         p12 = sunw_PKCS12_create(WANBOOT_PASSPHRASE, klist, clist, ta);
561         if (p12 == NULL) {
562                 wbku_printerr("cannot create %s: %s\n", name, cryptoerr());
563                 ret = -1;
564                 goto cleanup;
565         }

567         if (i2d_PKCS12_fp(fp, p12) == 0) {
568                 wbku_printerr("cannot write %s: %s\n", name, cryptoerr());
569                 ret = -1;
570                 goto cleanup;
571         }

573 cleanup:
574         (void) fclose(fp);
575         if (p12 != NULL)
576                 PKCS12_free(p12);
```

```
577         /*
578          * Put the cert and pkey off of the stack so that they won't
579          * be freed two times.  (If they get left in the stack then
580          * they will be freed with the stack.)
581          */
582         if (clist != NULL) {
583                 if (cert != NULL && sk_X509_num(clist) == 1) {
589                         /* LINTED */
584                         (void) sk_X509_delete(clist, 0);
585                 }
586                 sk_X509_pop_free(clist, X509_free);
587         }
588         if (klist != NULL) {
589                 if (pkey != NULL && sk_EVP_PKEY_num(klist) == 1) {
596                         /* LINTED */
590                         (void) sk_EVP_PKEY_delete(klist, 0);
591                 }
592                 sk_EVP_PKEY_pop_free(klist, sunw_evp_pkey_free);
593         }

595         return (ret);
596 }
```

unchanged_portion_omitted_

```
**********************************************************
    1236 Fri May 16 10:29:50 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/wanboot-cgi/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

include ../Makefile.com

PROG =          wanboot-cgi
LDLIBS +=       -lgen -lnsl -lwanbootutil -lnvpair -lwanboot
LDLIBS +=       -lgen -lnsl -lwanbootutil -lnvpair -lwanboot -lcrypto
CPPFLAGS +=     -I$(CMNCRYPTDIR)

# libcrypto has no lint library, so we can only include this while building
$(PROG) := LDLIBS += -lcrypto
#endif /* ! codereview */

all:            $(PROG)

install:        all $(ROOTCMD)

clean:

lint:           lint_PROG

include ../../../../Makefile.targ
```

```
**********************************************************
   45017 Fri May 16 10:29:51 2014
new/usr/src/cmd/cmd-inet/usr.lib/wanboot/wanboot-cgi/wanboot-cgi.c
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
______unchanged_portion_omitted_

 773 /*
 774  * Add the certs found in the trustfile found in path (a trust store) to
 775  * the file found at bootfs_dir/truststore.  If necessary, create the
 776  * output file.
 777  */
 778 static int
 779 build_trustfile(const char *path, void *truststorepath)
 780 {
 781         int             ret = WBCGI_FTW_CBERR;
 782         STACK_OF(X509)  *i_anchors = NULL;
 783         STACK_OF(X509)  *o_anchors = NULL;
 784         char            message[WBCGI_MAXBUF];
 785         PKCS12          *p12 = NULL;
 786         FILE            *rfp = NULL;
 787         FILE            *wfp = NULL;
 788         struct stat     i_st;
 789         struct stat     o_st;
 790         X509            *x = NULL;
 791         int             errtype = 0;
 792         int             wfd = -1;
 793         int             chars;
 794         int             i;

 796         if (!WBCGI_FILE_EXISTS(path, i_st)) {
 797                 goto cleanup;
 798         }

 800         if (WBCGI_FILE_EXISTS((char *)truststorepath, o_st)) {
 801                 /*
 802                  * If we are inadvertantly writing to the input file.
 803                  * return success.
 804                  * XXX Pete: how can this happen, and why success?
 805                  */
 806                 if (i_st.st_ino == o_st.st_ino) {
 807                         ret = WBCGI_FTW_CBCONT;
 808                         goto cleanup;
 809                 }
 810                 if ((wfp = fopen((char *)truststorepath, "r+")) == NULL) {
 811                         goto cleanup;
 812                 }
 813                 /*
 814                  * Read what's already there, so that new information
 815                  * can be added.
 816                  */
 817                 if ((p12 = d2i_PKCS12_fp(wfp, NULL)) == NULL) {
 818                         errtype = 1;
 819                         goto cleanup;
 820                 }
 821                 i = sunw_PKCS12_parse(p12, WANBOOT_PASSPHRASE, DO_NONE, NULL,
 822                     0, NULL, NULL, NULL, &o_anchors);
 823                 if (i <= 0) {
 824                         errtype = 1;
 825                         goto cleanup;
 826                 }

 828                 PKCS12_free(p12);
 829                 p12 = NULL;
```



```
 830         } else {
 831                 if (errno != ENOENT) {
 832                         chars = snprintf(message, sizeof (message),
 833                             "(error accessing file %s, error %s)",
 834                             path, strerror(errno));
 835                         if (chars > 0 && chars < sizeof (message))
 836                                 print_status(500, message);
 837                         else
 838                                 print_status(500, NULL);
 839                         return (WBCGI_FTW_CBERR);
 840                 }

 842                 /*
 843                  * Note: We could copy the file to the new trustfile, but
 844                  * we can't verify the password that way.  Therefore, copy
 845                  * it by reading it.
 846                  */
 847                 if ((wfd = open((char *)truststorepath,
 848                     O_CREAT|O_EXCL|O_RDWR, 0700)) < 0) {
 849                         goto cleanup;
 850                 }
 851                 if ((wfp = fdopen(wfd, "w+")) == NULL) {
 852                         goto cleanup;
 853                 }
 854                 o_anchors = sk_X509_new_null();
 855                 if (o_anchors == NULL) {
 856                         goto cleanup;
 857                 }
 858         }

 860         if ((rfp = fopen(path, "r")) == NULL) {
 861                 goto cleanup;
 862         }
 863         if ((p12 = d2i_PKCS12_fp(rfp, NULL)) == NULL) {
 864                 errtype = 1;
 865                 goto cleanup;
 866         }
 867         i = sunw_PKCS12_parse(p12, WANBOOT_PASSPHRASE, DO_NONE, NULL, 0, NULL,
 868             NULL, NULL, &i_anchors);
 869         if (i <= 0) {
 870                 errtype = 1;
 871                 goto cleanup;
 872         }
 873         PKCS12_free(p12);
 874         p12 = NULL;

 876         /*
 877          * Merge the two stacks of pkcs12 certs.
 878          */
 879         for (i = 0; i < sk_X509_num(i_anchors); i++) {
 880                 /* LINTED */
 880                 x = sk_X509_delete(i_anchors, i);
 881                 (void) sk_X509_push(o_anchors, x);
 882         }

 884         /*
 885          * Create the pkcs12 structure from the modified input stack and
 886          * then write out that structure.
 887          */
 888         p12 = sunw_PKCS12_create((const char *)WANBOOT_PASSPHRASE, NULL, NULL,
 889             o_anchors);
 890         if (p12 == NULL) {
 891                 goto cleanup;
 892         }
 893         rewind(wfp);
 894         if (i2d_PKCS12_fp(wfp, p12) == 0) {
```

```
 895                 goto cleanup;
 896         }

 898         ret = WBCGI_FTW_CBCONT;
 899 cleanup:
 900         if (ret == WBCGI_FTW_CBERR) {
 901                 if (errtype == 1) {
 902                         chars = snprintf(message, sizeof (message),
 903                             "(internal PKCS12 error while copying %s to %s)",
 904                             path, (char *)truststorepath);
 905                 } else {
 906                         chars = snprintf(message, sizeof (message),
 907                             "(error copying %s to %s)",
 908                             path, (char *)truststorepath);
 909                 }
 910                 if (chars > 0 && chars <= sizeof (message)) {
 911                         print_status(500, message);
 912                 } else {
 913                         print_status(500, NULL);
 914                 }
 915         }
 916         if (rfp != NULL) {
 917                 (void) fclose(rfp);
 918         }
 919         if (wfp != NULL) {
 920                 /* Will also close wfd */
 921                 (void) fclose(wfp);
 922         }
 923         if (p12 != NULL) {
 924                 PKCS12_free(p12);
 925         }
 926         if (i_anchors != NULL) {
 927                 sk_X509_pop_free(i_anchors, X509_free);
 928         }
 929         if (o_anchors != NULL) {
 930                 sk_X509_pop_free(o_anchors, X509_free);
 931         }

 933         return (ret);
 934 }
______unchanged_portion_omitted_

1056 /*
1057  * Loop through the certificates in a file
1058  */
1059 static int
1060 get_hostnames(const char *path, void *nvl)
1061 {
1062         int             ret = WBCGI_FTW_CBERR;
1063         STACK_OF(X509)  *certs = NULL;
1064         PKCS12          *p12 = NULL;
1065         char            message[WBCGI_MAXBUF];
1066         char            buf[WBCGI_MAXBUF + 1];
1067         FILE            *rfp = NULL;
1068         X509            *x = NULL;
1069         int             errtype = 0;
1070         int             chars;
1071         int             i;

1073         if ((rfp = fopen(path, "r")) == NULL) {
1074                 goto cleanup;
1075         }

1077         if ((p12 = d2i_PKCS12_fp(rfp, NULL)) == NULL) {
1078                 errtype = 1;
1079                 goto cleanup;
```



```
1080         }
1081         i = sunw_PKCS12_parse(p12, WANBOOT_PASSPHRASE, DO_NONE, NULL, 0, NULL,
1082             NULL, NULL, &certs);
1083         if (i <= 0) {
1084                 errtype = 1;
1085                 goto cleanup;
1086         }

1088         PKCS12_free(p12);
1089         p12 = NULL;

1091         for (i = 0; i < sk_X509_num(certs); i++) {
1093                 /* LINTED */
1092                 x = sk_X509_value(certs, i);
1093                 if (!one_name(sunw_issuer_attrs(x, buf, sizeof (buf) - 1),
1094                     nvl)) {
1095                         goto cleanup;
1096                 }
1097         }

1099         ret = WBCGI_FTW_CBCONT;
1100 cleanup:
1101         if (ret == WBCGI_FTW_CBERR) {
1102                 if (errtype == 1) {
1103                         chars = snprintf(message, sizeof (message),
1104                             "(internal PKCS12 error reading %s)", path);
1105                 } else {
1106                         chars = snprintf(message, sizeof (message),
1107                             "error reading %s", path);
1108                 }
1109                 if (chars > 0 && chars <= sizeof (message)) {
1110                         print_status(500, message);
1111                 } else {
1112                         print_status(500, NULL);
1113                 }
1114         }
1115         if (rfp != NULL) {
1116                 (void) fclose(rfp);
1117         }
1118         if (p12 != NULL) {
1119                 PKCS12_free(p12);
1120         }
1121         if (certs != NULL) {
1122                 sk_X509_pop_free(certs, X509_free);
1123         }

1125         return (ret);
1126 }
______unchanged_portion_omitted_
```

```
**********************************************************
    2444 Fri May 16 10:29:51 2014
new/usr/src/cmd/ssh/libssh/Makefile.com
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBRARY =       libssh.a
VERS =          .1

OBJECTS =       \
                addrmatch.o \
                atomicio.o \
                authfd.o \
                authfile.o \
                bufaux.o \
                buffer.o \
                canohost.o \
                channels.o \
                cipher.o \
                cipher-ctr.o \
                compat.o \
                compress.o \
                crc32.o \
                deattack.o \
                dh.o \
                dispatch.o \
                engine.o \
                entropy.o \
                fatal.o \
                g11n.o \
                hostfile.o \
                key.o \
                kex.o \
                kexdh.o \
                kexdhc.o \
                kexdhs.o \
                kexgex.o \
                kexgexc.o \
                kexgexs.o \
                kexgssc.o \
                kexgsss.o \
```



```
                log.o \
                mac.o \
                match.o \
                misc.o \
                mpaux.o \
                msg.o \
                nchan.o \
                packet.o \
                progressmeter.o \
                proxy-io.o \
                radix.o \
                readconf.o \
                readpass.o \
                rsa.o \
                sftp-common.o \
                ssh-dss.o \
                ssh-gss.o \
                ssh-rsa.o \
                tildexpand.o \
                ttymodes.o \
                uidswap.o \
                uuencode.o \
                xlist.o \
                xmalloc.o

include $(SRC)/lib/Makefile.lib

BUILD.AR =       $(RM) $@ ; $(AR) $(ARFLAGS) $@ $(AROBJS)

SRCDIR =        ../common
SRCS =          $(OBJECTS:%.o=../common/%.c)

LIBS =          $(LIBRARY) $(LINTLIB)

# Define LDLIBS conditionally for lintcheck, rather than in general, since
# we're building an archive library which itself links to nothing, we just
# want lint to know about these libraries.
lintcheck  :=   LDLIBS += -lz -lsocket -lnsl -lc
lintcheck  :=   LDLIBS += -lcrypto -lz -lsocket -lnsl -lc
$(LINTLIB) :=   SRCS = $(SRCDIR)/$(LINTSRC)

POFILE_DIR =    ../..

.KEEP_STATE:

all:            $(LIBS)

# lint requires the (not installed) lint library
lint:           $(LINTLIB) .WAIT lintcheck

include $(SRC)/lib/Makefile.targ

objs/%.o:       $(SRCDIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

include ../../Makefile.ssh-common
include ../../Makefile.msg.targ
```

```
**********************************************************
    1595 Fri May 16 10:29:51 2014
new/usr/src/cmd/ssh/sftp-server/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #
  24 # cmd/ssh/sftp-server/Makefile

  26 PROG =          sftp-server

  28 OBJS =          sftp-server.o sftp-server-main.o
  29 SRCS =          $(OBJS:.o=.c)

  31 include ../../Makefile.cmd
  32 include ../Makefile.ssh-common

  34 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket

  36 # libcrypto has no lint library, so we can only use it when building
  37 $(PROG) :=      LDLIBS += -lcrypto
  34 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket -lcrypto

  39 POFILE_DIR =    ..

  41 .KEEP_STATE:

  43 .PARALLEL:      $(OBJS)

  45 all:            $(PROG)

  47 $(PROG):        $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/
  48         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
  49         $(POST_PROCESS)

  51 install:        all $(ROOTLIBSSHPROG) $(ROOTLIBSSH)

  53 clean:
  54         $(RM) -f $(OBJS) $(PROG)

  56 lint:           lint_SRCS

  58 include ../Makefile.msg.targ
```



```
  59 include ../../Makefile.targ
```

```
**********************************************************
    1580 Fri May 16 10:29:51 2014
new/usr/src/cmd/ssh/sftp/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #
  24 # cmd/ssh/sftp/Makefile

  26 PROG =          sftp

  28 OBJS =          \
  29                 sftp.o \
  30                 sftp-client.o \
  31                 sftp-glob.o

  33 SRCS =          $(OBJS:.o=.c)

  35 include ../../Makefile.cmd
  36 include ../Makefile.ssh-common

  38 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket -ltecla

  40 # libcrypto has no lint library, so we can only use it when building
  41 $(PROG) := LDLIBS += -lcrypto
  38 LDLIBS +=       $(SSH_COMMON_LDLIBS) -lsocket -lcrypto -ltecla

  43 POFILE_DIR =    ..

  45 .KEEP_STATE:

  47 .PARALLEL:      $(OBJS)

  49 all: $(PROG)

  51 $(PROG):        $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/
  52         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
  53         $(POST_PROCESS)

  55 install:        all $(ROOTPROG)

  57 clean:
  58         $(RM) -f $(OBJS) $(PROG)
```



```
  60 lint:           lint_SRCS

  62 include ../Makefile.msg.targ
  63 include ../../Makefile.targ
```

```
**********************************************************
    1542 Fri May 16 10:29:52 2014
new/usr/src/cmd/ssh/ssh-add/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/ssh/ssh-add/Makefile

PROG= ssh-add

OBJS    =       \
        ssh-add.o
SRCS    = $(OBJS:.o=.c)

include ../../Makefile.cmd
include ../Makefile.ssh-common

LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket

# libcrypto has no lint library, so we can only use it when building
$(PROG) := LDLIBS += -lcrypto
LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lcrypto

POFILE_DIR= ..

.KEEP_STATE:

.PARALLEL: $(OBJS)

all: $(PROG)

$(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
        $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
        $(POST_PROCESS)

install: all $(ROOTPROG)

clean:
        $(RM) -f $(OBJS) $(PROG)

lint:   lint_SRCS
```



```
include ../Makefile.msg.targ
include ../../Makefile.targ
```

```
**********************************************************
    1548 Fri May 16 10:29:52 2014
new/usr/src/cmd/ssh/ssh-agent/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #
  24 # cmd/ssh/ssh-agent/Makefile

  26 PROG= ssh-agent

  28 OBJS    =       \
  29         ssh-agent.o
  30 SRCS    = $(OBJS:.o=.c)

  32 include ../../Makefile.cmd
  33 include ../Makefile.ssh-common

  35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket

  37 # libcrypto has no lint library, so we can only use it when building
  38 $(PROG) := LDLIBS += -lcrypto
  35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lcrypto

  40 POFILE_DIR= ..

  42 .KEEP_STATE:

  44 .PARALLEL: $(OBJS)

  46 all: $(PROG)

  48 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
  49         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
  50         $(POST_PROCESS)

  52 install: all $(ROOTPROG)

  54 clean:
  55         $(RM) -f $(OBJS) $(PROG)

  57 lint:   lint_SRCS
```



```
  59 include ../Makefile.msg.targ
  60 include ../../Makefile.targ
```

```
**********************************************************
    1551 Fri May 16 10:29:52 2014
new/usr/src/cmd/ssh/ssh-keygen/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/ssh/ssh-keygen/Makefile

PROG= ssh-keygen

OBJS    =       \
        ssh-keygen.o
SRCS    = $(OBJS:.o=.c)

include ../../Makefile.cmd
include ../Makefile.ssh-common

LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket

# libcrypto has no lint library, so we can only use it when building
$(PROG) := LDLIBS += -lcrypto
LDLIBS += $(SSH_COMMON_LDLIBS) -lcrypto -lsocket

POFILE_DIR= ..

.KEEP_STATE:

.PARALLEL: $(OBJS)

all: $(PROG)

$(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
        $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
        $(POST_PROCESS)

install: all $(ROOTPROG)

clean:
        $(RM) -f $(OBJS) $(PROG)

lint:   lint_SRCS
```



```
include ../Makefile.msg.targ
include ../../Makefile.targ
```

```
**********************************************************
    1564 Fri May 16 10:29:52 2014
new/usr/src/cmd/ssh/ssh-keyscan/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #
  24 # cmd/ssh/ssh-keyscan/Makefile

  26 PROG= ssh-keyscan

  28 OBJS    =       \
  29         ssh-keyscan.o
  30 SRCS    = $(OBJS:.o=.c)

  32 include ../../Makefile.cmd
  33 include ../Makefile.ssh-common

  35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lnsl -lz

  37 # libcrypto has no lint library, so we can only use it when building
  38 $(PROG) := LDLIBS += -lcrypto
  35 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lnsl -lz -lcrypto

  40 POFILE_DIR= ..

  42 .KEEP_STATE:

  44 .PARALLEL: $(OBJS)

  46 all: $(PROG)

  48 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
  49         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
  50         $(POST_PROCESS)

  52 clean:
  53         $(RM) -f $(OBJS) $(PROG)

  55 lint:   lint_SRCS

  57 include ../Makefile.msg.targ
  58 include ../../Makefile.targ
```



```
  60 install: all $(ROOTPROG)
```

```
**********************************************************
    1685 Fri May 16 10:29:53 2014
new/usr/src/cmd/ssh/ssh-keysign/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #
  21 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
  22 # Use is subject to license terms.
  23 #
  24 # cmd/ssh/ssh-keysign/Makefile

  26 PROG= ssh-keysign

  28 DIRS= $(ROOTLIBSSH)


  31 OBJS    = ssh-keysign.o
  32 SRCS    = $(OBJS:.o=.c)

  34 include ../../Makefile.cmd
  35 include ../Makefile.ssh-common

  37 FILEMODE= 04555

  39 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lnsl -lz

  41 # libcrypto has no lint library, so we can only use it when building
  42 $(PROG) := LDLIBS += -lcrypto
  39 LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket -lnsl -lz -lcrypto

  44 POFILE_DIR= ..

  46 .KEEP_STATE:

  48 .PARALLEL: $(OBJS)

  50 all: $(PROG)

  52 $(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
  53         $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
  54         $(POST_PROCESS)

  56 clean:
  57         $(RM) -f $(OBJS) $(PROG)
```



```
  59 lint:   lint_SRCS

  61 include ../Makefile.msg.targ
  62 include ../../Makefile.targ

  64 install: all $(DIRS) $(ROOTLIBSSHPROG) $(ROOTLIBSSH)


  67 $(ROOTLIBSSHPROG)/%: %
  68         $(INS.file)

  70 $(DIRS):
  71         $(INS.dir)
```

```
**********************************************************
    1681 Fri May 16 10:29:53 2014
new/usr/src/cmd/ssh/ssh/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/ssh/ssh/Makefile

PROG= ssh

OBJS    = ssh.o \
          sshconnect.o \
          sshconnect1.o \
          sshconnect2.o \
          sshtty.o \
          clientloop.o \
          gss-clnt.o
SRCS    = $(OBJS:.o=.c)

include ../../Makefile.cmd
include ../Makefile.ssh-common

LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket \
        -lnsl \
        -lz \
        -lcrypto \
        -lgss

# libcrypto has no lint library, so we can only use it when building
$(PROG) := LDLIBS += -lcrypto
#endif /* ! codereview */

POFILE_DIR= ..

.KEEP_STATE:

.PARALLEL: $(OBJS)

all: $(PROG)

$(PROG): $(OBJS) ../libssh/$(MACH)/libssh.a ../libopenbsd-compat/$(MACH)/libopen
        $(LINK.c) $(OBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
```



```
        $(POST_PROCESS)

install: all $(ROOTPROG)

clean:
        $(RM) -f $(OBJS) $(PROG)

lint:   lint_SRCS

include ../Makefile.msg.targ

XGETFLAGS += --keyword=log
include ../../Makefile.targ
```

```
**********************************************************
    2503 Fri May 16 10:29:53 2014
new/usr/src/cmd/ssh/sshd/Makefile
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# cmd/ssh/sshd/Makefile

PROG= sshd

DIRS= $(ROOTLIBSSH)

OBJS    = sshd.o \
          altprivsep.o \
          auth.o \
          auth1.o \
          auth2.o \
          auth-options.o \
          auth2-chall.o \
          auth2-gss.o \
          auth2-hostbased.o \
          auth2-kbdint.o \
          auth2-none.o \
          auth2-passwd.o \
          auth2-pam.o \
          auth2-pubkey.o \
          auth-bsdauth.o \
          auth-chall.o \
          auth-rhosts.o \
          auth-krb4.o \
          auth-krb5.o \
          auth-pam.o \
          auth-passwd.o \
          auth-rsa.o \
          auth-rh-rsa.o \
          auth-sia.o \
          auth-skey.o \
          bsmaudit.o \
          groupaccess.o \
          gss-serv.o \
          loginrec.o \
          servconf.o \
```



```
          serverloop.o \
          session.o \
          sshlogin.o \
          sshpty.o

EXTOBJS = sftp-server.o

SRCS    = $(OBJS:.o=.c) ../sftp-server/sftp-server.c

include ../../Makefile.cmd
include ../Makefile.ssh-common

LDLIBS += $(SSH_COMMON_LDLIBS) -lsocket \
        -lnsl \
        -lz \
        -lpam \
        -lbsm \
        -lwrap \
        -lcrypto \
        -lgss \
        -lcontract
MAPFILES = $(MAPFILE.INT) $(MAPFILE.NGB)
LDFLAGS += $(MAPFILES:%=-M%)

# libcrypto has no lint library, so we can only use it when building
$(PROG) := LDLIBS += -lcrypto
#endif /* ! codereview */

POFILE_DIR= ..

.KEEP_STATE:

.PARALLEL: $(OBJS)

all: $(PROG)

$(PROG): $(OBJS) $(EXTOBJS) $(MAPFILES) ../libssh/$(MACH)/libssh.a \
        ../libopenbsd-compat/$(MACH)/libopenbsd-compat.a
        $(LINK.c) $(OBJS) $(EXTOBJS) -o $@ $(LDLIBS) $(DYNFLAGS)
        $(POST_PROCESS)

%.o : ../sftp-server/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

install: all $(DIRS) $(ROOTLIBSSHPROG) $(ROOTLIBSSH)


$(ROOTLIBSSHPROG)/%: %
        $(INS.file)

$(DIRS):
        $(INS.dir)

clean:
        $(RM) $(OBJS) $(EXTOBJS)

lint:   lint_SRCS

include ../Makefile.msg.targ
include ../../Makefile.targ
```

```
**********************************************************
   72083 Fri May 16 10:29:53 2014
new/usr/src/common/net/wanboot/boot_http.c
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
______unchanged_portion_omitted_

2230 /*
2231  * print_ciphers - Print the list of ciphers for debugging.
2232  *
2233  *       print_ciphers(ssl);
2234  *
2235  * Arguments:
2236  *     ssl      - SSL connection.
2237  *
2238  * Returns:
2239  *     none
2240  */
2241 static void
2242 print_ciphers(SSL *ssl)
2243 {
2244         SSL_CIPHER      *c;
2245         STACK_OF(SSL_CIPHER)    *sk;
2246         int     i;
2247         const char      *name;

2249         if (ssl == NULL)
2250                 return;

2252         sk = SSL_get_ciphers(ssl);
2253         if (sk == NULL)
2254                 return;

2256         for (i = 0; i < sk_SSL_CIPHER_num(sk); i++) {
2257                 /* LINTED */
2257                 c = sk_SSL_CIPHER_value(sk, i);
2258                 libbootlog(BOOTLOG_VERBOSE, "%08lx %s", c->id, c->name);
2259         }
2260         name = SSL_get_cipher_name(ssl);
2261         if (name == NULL)
2262                 name = "";
2263         libbootlog(BOOTLOG_VERBOSE, "Current cipher = %s", name);
2264 }
______unchanged_portion_omitted_
```

```
**********************************************************
    1939 Fri May 16 10:29:54 2014
new/usr/src/lib/libkmf/plugins/kmf_openssl/Makefile.com
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Makefile for KMF Plugins
#

LIBRARY=        kmf_openssl.a
VERS=           .1

OBJECTS=        openssl_spi.o

include $(SRC)/lib/Makefile.lib

LIBLINKS=       $(DYNLIB:.so.1=.so)
KMFINC=         -I../../../include -I../../../ber_der/inc

BERLIB=         -lkmf -lkmfberder
BERLIB64=       $(BERLIB)

OPENSSLLIBS=    $(BERLIB) -lcrypto -lcryptoutil -lc
OPENSSLLIBS64=  $(BERLIB64) -lcrypto -lcryptoutil -lc

LINTSSLLIBS     = $(BERLIB) -lcryptoutil -lc
LINTSSLLIBS64   = $(BERLIB64) -lcryptoutil -lc
LINTSSLLIBS     = $(BERLIB) -lcrypto -lcryptoutil -lc
LINTSSLLIBS64   = $(BERLIB64) -lcrypto -lcryptoutil -lc

SRCDIR=         ../common
INCDIR=         ../../include

CFLAGS          +=      $(CCVERBOSE)
CPPFLAGS        +=      -D_REENTRANT $(KMFINC) \
                        -I$(INCDIR) -I$(ADJUNCT_PROTO)/usr/include/libxml2

CERRWARN        +=      -_gcc=-Wno-unused-label
CERRWARN        +=      -_gcc=-Wno-unused-value
CERRWARN        +=      -_gcc=-Wno-uninitialized

PICS=   $(OBJECTS:%=pics/%)
```



```
lint:=  OPENSSLLIBS=    $(LINTSSLLIBS)
lint:=  OPENSSLLIBS64=  $(LINTSSLLIBS64)

LDLIBS32        +=      $(OPENSSLLIBS)

ROOTLIBDIR=     $(ROOTFS_LIBDIR)/crypto
ROOTLIBDIR64=   $(ROOTFS_LIBDIR)/crypto/$(MACH64)

.KEEP_STATE:

LIBS    =       $(DYNLIB)
all:    $(DYNLIB) $(LINTLIB)

lint: lintcheck

FRC:

include $(SRC)/lib/Makefile.targ
```

```
**********************************************************
  133359 Fri May 16 10:29:54 2014
new/usr/src/lib/libkmf/plugins/kmf_openssl/common/openssl_spi.c
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
______unchanged_portion_omitted_

2483 /* ocsp_find_signer_sk() is copied from openssl source */
2484 static X509 *ocsp_find_signer_sk(STACK_OF(X509) *certs, OCSP_RESPID *id)
2485 {
2486         int i;
2487         unsigned char tmphash[SHA_DIGEST_LENGTH], *keyhash;

2489         /* Easy if lookup by name */
2490         if (id->type == V_OCSP_RESPID_NAME)
2491                 return (X509_find_by_subject(certs, id->value.byName));

2493         /* Lookup by key hash */

2495         /* If key hash isn't SHA1 length then forget it */
2496         if (id->value.byKey->length != SHA_DIGEST_LENGTH)
2497                 return (NULL);

2499         keyhash = id->value.byKey->data;
2500         /* Calculate hash of each key and compare */
2501         for (i = 0; i < sk_X509_num(certs); i++) {
2502                 /* LINTED E_BAD_PTR_CAST_ALIGN */
2502                 X509 *x = sk_X509_value(certs, i);
2503                 /* Use pubkey_digest to get the key ID value */
2504                 (void) X509_pubkey_digest(x, EVP_sha1(), tmphash, NULL);
2505                 if (!memcmp(keyhash, tmphash, SHA_DIGEST_LENGTH))
2506                         return (x);
2507         }
2508         return (NULL);
2509 }
______unchanged_portion_omitted_

3595 /*
3596  * Helper function to extract keys and certificates from
3597  * a single PEM file.  Typically the file should contain a
3598  * private key and an associated public key wrapped in an x509 cert.
3599  * However, the file may be just a list of X509 certs with no keys.
3600  */
3601 static KMF_RETURN
3602 extract_pem(KMF_HANDLE *kmfh,
3603         char *issuer, char *subject, KMF_BIGINT *serial,
3604         char *filename, CK_UTF8CHAR *pin,
3605         CK_ULONG pinlen, EVP_PKEY **priv_key, KMF_DATA **certs,
3606         int *numcerts)
3607 /* ARGSUSED6 */
3608 {
3609         KMF_RETURN rv = KMF_OK;
3610         FILE *fp;
3611         STACK_OF(X509_INFO) *x509_info_stack = NULL;
3612         int i, ncerts = 0, matchcerts = 0;
3613         EVP_PKEY *pkey = NULL;
3614         X509_INFO *info;
3615         X509 *x;
3616         X509_INFO **cert_infos = NULL;
3617         KMF_DATA *certlist = NULL;

3619         if (priv_key)
3620                 *priv_key = NULL;
3621         if (certs)
```



```
3622                 *certs = NULL;
3623         fp = fopen(filename, "r");
3624         if (fp == NULL)
3625                 return (KMF_ERR_OPEN_FILE);

3627         x509_info_stack = PEM_X509_INFO_read(fp, NULL, NULL, pin);
3628         if (x509_info_stack == NULL) {
3629                 (void) fclose(fp);
3630                 return (KMF_ERR_ENCODING);
3631         }
3632         cert_infos = (X509_INFO **)malloc(sk_X509_INFO_num(x509_info_stack) *
3633             sizeof (X509_INFO *));
3634         if (cert_infos == NULL) {
3635                 (void) fclose(fp);
3636                 rv = KMF_ERR_MEMORY;
3637                 goto err;
3638         }

3640         for (i = 0; i < sk_X509_INFO_num(x509_info_stack); i++) {
3642                 /* LINTED E_BAD_PTR_CAST_ALIGN */
3641                 cert_infos[ncerts] = sk_X509_INFO_value(x509_info_stack, i);
3642                 ncerts++;
3643         }

3645         if (ncerts == 0) {
3646                 (void) fclose(fp);
3647                 rv = KMF_ERR_CERT_NOT_FOUND;
3648                 goto err;
3649         }

3651         if (priv_key != NULL) {
3652                 rewind(fp);
3653                 pkey = PEM_read_PrivateKey(fp, NULL, NULL, pin);
3654         }
3655         (void) fclose(fp);

3657         x = cert_infos[ncerts - 1]->x509;
3658         /*
3659          * Make sure the private key matchs the last cert in the file.
3660          */
3661         if (pkey != NULL && !X509_check_private_key(x, pkey)) {
3662                 EVP_PKEY_free(pkey);
3663                 rv = KMF_ERR_KEY_MISMATCH;
3664                 goto err;
3665         }

3667         certlist = (KMF_DATA *)calloc(ncerts, sizeof (KMF_DATA));
3668         if (certlist == NULL) {
3669                 if (pkey != NULL)
3670                         EVP_PKEY_free(pkey);
3671                 rv = KMF_ERR_MEMORY;
3672                 goto err;
3673         }

3675         /*
3676          * Convert all of the certs to DER format.
3677          */
3678         matchcerts = 0;
3679         for (i = 0; rv == KMF_OK && certs != NULL && i < ncerts; i++) {
3680                 boolean_t match = FALSE;
3681                 info =  cert_infos[ncerts - 1 - i];

3683                 rv = check_cert(info->x509, issuer, subject, serial, &match);
3684                 if (rv != KMF_OK || match != TRUE) {
3685                         rv = KMF_OK;
3686                         continue;
```

```
3687                }

3689                rv = ssl_cert2KMFDATA(kmfh, info->x509,
3690                        &certlist[matchcerts++]);

3692                if (rv != KMF_OK) {
3693                        int j;
3694                        for (j = 0; j < matchcerts; j++)
3695                                kmf_free_data(&certlist[j]);
3696                        free(certlist);
3697                        certlist = NULL;
3698                        ncerts = matchcerts = 0;
3699                }
3700        }

3702        if (numcerts != NULL)
3703                *numcerts = matchcerts;

3705        if (certs != NULL)
3706                *certs = certlist;
3707        else if (certlist != NULL) {
3708                for (i = 0; i < ncerts; i++)
3709                        kmf_free_data(&certlist[i]);
3710                free(certlist);
3711                certlist = NULL;
3712        }

3714        if (priv_key == NULL && pkey != NULL)
3715                EVP_PKEY_free(pkey);
3716        else if (priv_key != NULL && pkey != NULL)
3717                *priv_key = pkey;

3719 err:
3720        /* Cleanup the stack of X509 info records */
3721        for (i = 0; i < sk_X509_INFO_num(x509_info_stack); i++) {
3724                /* LINTED E_BAD_PTR_CAST_ALIGN */
3722                info = (X509_INFO *)sk_X509_INFO_value(x509_info_stack, i);
3723                X509_INFO_free(info);
3724        }
3725        if (x509_info_stack)
3726                sk_X509_INFO_free(x509_info_stack);

3728        if (cert_infos != NULL)
3729                free(cert_infos);

3731        return (rv);
3732 }

3734 static KMF_RETURN
3735 openssl_parse_bags(STACK_OF(PKCS12_SAFEBAG) *bags, char *pin,
3736        STACK_OF(EVP_PKEY) *keys, STACK_OF(X509) *certs)
3737 {
3738        KMF_RETURN ret;
3739        int i;

3741        for (i = 0; i < sk_PKCS12_SAFEBAG_num(bags); i++) {
3745                /* LINTED E_BAD_PTR_CAST_ALIGN */
3742                PKCS12_SAFEBAG *bag = sk_PKCS12_SAFEBAG_value(bags, i);
3743                ret = openssl_parse_bag(bag, pin, (pin ? strlen(pin) : 0),
3744                    keys, certs);

3746                if (ret != KMF_OK)
3747                        return (ret);
3748        }

3750        return (ret);
```



```
3751 }

3753 static KMF_RETURN
3754 set_pkey_attrib(EVP_PKEY *pkey, ASN1_TYPE *attrib, int nid)
3755 {
3756        X509_ATTRIBUTE *attr = NULL;

3758        if (pkey == NULL || attrib == NULL)
3759                return (KMF_ERR_BAD_PARAMETER);

3761        if (pkey->attributes == NULL) {
3762                pkey->attributes = sk_X509_ATTRIBUTE_new_null();
3763                if (pkey->attributes == NULL)
3764                        return (KMF_ERR_MEMORY);
3765        }
3766        attr = X509_ATTRIBUTE_create(nid, attrib->type, attrib->value.ptr);
3767        if (attr != NULL) {
3768                int i;
3769                X509_ATTRIBUTE *a;
3770                for (i = 0;
3771                    i < sk_X509_ATTRIBUTE_num(pkey->attributes); i++) {
3776                        /* LINTED E_BAD_PTR_CASE_ALIGN */
3772                        a = sk_X509_ATTRIBUTE_value(pkey->attributes, i);
3773                        if (OBJ_obj2nid(a->object) == nid) {
3774                                X509_ATTRIBUTE_free(a);
3775                                (void) sk_X509_ATTRIBUTE_set(pkey->attributes,
3780                                /* LINTED E_BAD_PTR_CAST_ALIGN */
3781                                sk_X509_ATTRIBUTE_set(pkey->attributes,
3776                                    i, attr);
3777                                return (KMF_OK);
3778                        }
3779                }
3780                if (sk_X509_ATTRIBUTE_push(pkey->attributes, attr) == NULL) {
3781                        X509_ATTRIBUTE_free(attr);
3782                        return (KMF_ERR_MEMORY);
3783                }
3784        } else {
3785                return (KMF_ERR_MEMORY);
3786        }

3788        return (KMF_OK);
3789 }
______unchanged_portion_omitted_

3919 static KMF_RETURN
3920 openssl_pkcs12_parse(PKCS12 *p12, char *pin,
3921        STACK_OF(EVP_PKEY) *keys,
3922        STACK_OF(X509) *certs,
3923        STACK_OF(X509) *ca)
3924 /* ARGSUSED3 */
3925 {
3926        KMF_RETURN ret = KMF_OK;
3927        STACK_OF(PKCS7) *asafes = NULL;
3928        STACK_OF(PKCS12_SAFEBAG) *bags = NULL;
3929        int i, bagnid;
3930        PKCS7 *p7;

3932        if (p12 == NULL || (keys == NULL && certs == NULL))
3933                return (KMF_ERR_BAD_PARAMETER);

3935        if (pin == NULL || *pin == NULL) {
3936                if (PKCS12_verify_mac(p12, NULL, 0)) {
3937                        pin = NULL;
3938                } else if (PKCS12_verify_mac(p12, "", 0)) {
3939                        pin = "";
3940                } else {
```

```
3941                        return (KMF_ERR_AUTH_FAILED);
3942                }
3943        } else if (!PKCS12_verify_mac(p12, pin, -1)) {
3944                return (KMF_ERR_AUTH_FAILED);
3945        }

3947        if ((asafes = PKCS12_unpack_authsafes(p12)) == NULL)
3948                return (KMF_ERR_PKCS12_FORMAT);

3950        for (i = 0; ret == KMF_OK && i < sk_PKCS7_num(asafes); i++) {
3951                bags = NULL;
3958                /* LINTED E_BAD_PTR_CAST_ALIGN */
3952                p7 = sk_PKCS7_value(asafes, i);
3953                bagnid = OBJ_obj2nid(p7->type);

3955                if (bagnid == NID_pkcs7_data) {
3956                        bags = PKCS12_unpack_p7data(p7);
3957                } else if (bagnid == NID_pkcs7_encrypted) {
3958                        bags = PKCS12_unpack_p7encdata(p7, pin,
3959                            (pin ? strlen(pin) : 0));
3960                } else {
3961                        continue;
3962                }
3963                if (bags == NULL) {
3964                        ret = KMF_ERR_PKCS12_FORMAT;
3965                        goto out;
3966                }

3968                if (openssl_parse_bags(bags, pin, keys, certs) != KMF_OK)
3969                        ret = KMF_ERR_PKCS12_FORMAT;

3971                sk_PKCS12_SAFEBAG_pop_free(bags, PKCS12_SAFEBAG_free);
3972        }
3973 out:
3974        if (asafes != NULL)
3975                sk_PKCS7_pop_free(asafes, PKCS7_free);

3977        return (ret);
3978 }
```

unchanged_portion_omitted_

```
4219 static X509_ATTRIBUTE *
4220 find_attr(STACK_OF(X509_ATTRIBUTE) *attrs, int nid)
4221 {
4222        X509_ATTRIBUTE *a;
4223        int i;

4225        if (attrs == NULL)
4226                return (NULL);

4228        for (i = 0; i < sk_X509_ATTRIBUTE_num(attrs); i++) {
4236                /* LINTED E_BAD_PTR_CAST_ALIGN */
4229                a = sk_X509_ATTRIBUTE_value(attrs, i);
4230                if (OBJ_obj2nid(a->object) == nid)
4231                        return (a);
4232        }
4233        return (NULL);
4234 }

4236 static KMF_RETURN
4237 convertToRawKey(EVP_PKEY *pkey, KMF_RAW_KEY_DATA *key)
4238 {
4239        KMF_RETURN rv = KMF_OK;
4240        X509_ATTRIBUTE *attr;

4242        if (pkey == NULL || key == NULL)
```



```
4243                return (KMF_ERR_BAD_PARAMETER);
4244        /* Convert SSL key to raw key */
4245        switch (pkey->type) {
4246                case EVP_PKEY_RSA:
4247                        rv = exportRawRSAKey(EVP_PKEY_get1_RSA(pkey),
4248                            key);
4249                        if (rv != KMF_OK)
4250                                return (rv);
4251                        break;
4252                case EVP_PKEY_DSA:
4253                        rv = exportRawDSAKey(EVP_PKEY_get1_DSA(pkey),
4254                            key);
4255                        if (rv != KMF_OK)
4256                                return (rv);
4257                        break;
4258                default:
4259                        return (KMF_ERR_BAD_PARAMETER);
4260        }
4261        /*
4262         * If friendlyName, add it to record.
4263         */
4264        attr = find_attr(pkey->attributes, NID_friendlyName);
4265        if (attr != NULL) {
4266                ASN1_TYPE *ty = NULL;
4267                int numattr = sk_ASN1_TYPE_num(attr->value.set);
4268                if (attr->single == 0 && numattr > 0) {
4277                        /* LINTED E_BAD_PTR_CAST_ALIGN */
4269                        ty = sk_ASN1_TYPE_value(attr->value.set, 0);
4270                }
4271                if (ty != NULL) {
4272 #if OPENSSL_VERSION_NUMBER < 0x10000000L
4273                        key->label = uni2asc(ty->value.bmpstring->data,
4274                            ty->value.bmpstring->length);
4275 #else
4276                        key->label = OPENSSL_uni2asc(ty->value.bmpstring->data,
4277                            ty->value.bmpstring->length);
4278 #endif
4279                }
4280        } else {
4281                key->label = NULL;
4282        }

4284        /*
4285         * If KeyID, add it to record as a KMF_DATA object.
4286         */
4287        attr = find_attr(pkey->attributes, NID_localKeyID);
4288        if (attr != NULL) {
4289                ASN1_TYPE *ty = NULL;
4290                int numattr = sk_ASN1_TYPE_num(attr->value.set);
4291                if (attr->single == 0 && numattr > 0) {
4301                        /* LINTED E_BAD_PTR_CAST_ALIGN */
4292                        ty = sk_ASN1_TYPE_value(attr->value.set, 0);
4293                }
4294                key->id.Data = (uchar_t *)malloc(
4295                    ty->value.octet_string->length);
4296                if (key->id.Data == NULL)
4297                        return (KMF_ERR_MEMORY);
4298                (void) memcpy(key->id.Data, ty->value.octet_string->data,
4299                    ty->value.octet_string->length);
4300                key->id.Length = ty->value.octet_string->length;
4301        } else {
4302                (void) memset(&key->id, 0, sizeof (KMF_DATA));
4303        }

4305        return (rv);
4306 }
```

```
4308 static KMF_RETURN
4309 convertPK12Objects(
4310         KMF_HANDLE *kmfh,
4311         STACK_OF(EVP_PKEY) *sslkeys,
4312         STACK_OF(X509) *sslcert,
4313         STACK_OF(X509) *sslcacerts,
4314         KMF_RAW_KEY_DATA **keylist, int *nkeys,
4315         KMF_X509_DER_CERT **certlist, int *ncerts)
4316 {
4317         KMF_RETURN rv = KMF_OK;
4318         KMF_RAW_KEY_DATA key;
4319         int i;

4321         for (i = 0; sslkeys != NULL && i < sk_EVP_PKEY_num(sslkeys); i++) {
4332                 /* LINTED E_BAD_PTR_CAST_ALIGN */
4322                 EVP_PKEY *pkey = sk_EVP_PKEY_value(sslkeys, i);
4323                 rv = convertToRawKey(pkey, &key);
4324                 if (rv == KMF_OK)
4325                         rv = add_key_to_list(keylist, &key, nkeys);

4327                 if (rv != KMF_OK)
4328                         return (rv);
4329         }

4331         /* Now add the certificate to the certlist */
4332         for (i = 0; sslcert != NULL && i < sk_X509_num(sslcert); i++) {
4344                 /* LINTED E_BAD_PTR_CAST_ALIGN */
4333                 X509 *cert = sk_X509_value(sslcert, i);
4334                 rv = add_cert_to_list(kmfh, cert, certlist, ncerts);
4335                 if (rv != KMF_OK)
4336                         return (rv);
4337         }

4339         /* Also add any included CA certs to the list */
4340         for (i = 0; sslcacerts != NULL && i < sk_X509_num(sslcacerts); i++) {
4341                 X509 *c;
4342                 /*
4343                  * sk_X509_value() is macro that embeds a cast to (X509 *).
4344                  * Here it translates into ((X509 *)sk_value((ca), (i))).
4345                  * Lint is complaining about the embedded casting, and
4346                  * to fix it, you need to fix openssl header files.
4347                  */
4360                 /* LINTED E_BAD_PTR_CAST_ALIGN */
4348                 c = sk_X509_value(sslcacerts, i);

4350                 /* Now add the ca cert to the certlist */
4351                 rv = add_cert_to_list(kmfh, c, certlist, ncerts);
4352                 if (rv != KMF_OK)
4353                         return (rv);
4354         }
4355         return (rv);
4356 }
```

unchanged_portion_omitted_

```
5309 KMF_RETURN
5310 OpenSSL_FindCertInCRL(KMF_HANDLE_T handle, int numattr, KMF_ATTRIBUTE *attrlist)
5311 {
5312         KMF_RETURN ret = KMF_OK;
5313         KMF_HANDLE *kmfh = (KMF_HANDLE *)handle;
5314         KMF_ENCODE_FORMAT format;
5315         BIO *in = NULL;
5316         X509   *xcert = NULL;
5317         X509_CRL   *xcrl = NULL;
5318         STACK_OF(X509_REVOKED) *revoke_stack = NULL;
5319         X509_REVOKED *revoke;
```



```
5320         int i;
5321         char *crlfilename, *crlfile, *dirpath, *certfile;

5323         if (numattr == 0 || attrlist == NULL) {
5324                 return (KMF_ERR_BAD_PARAMETER);
5325         }

5327         crlfilename = kmf_get_attr_ptr(KMF_CRL_FILENAME_ATTR,
5328             attrlist, numattr);

5330         if (crlfilename == NULL)
5331                 return (KMF_ERR_BAD_CRLFILE);

5333         certfile = kmf_get_attr_ptr(KMF_CERT_FILENAME_ATTR, attrlist, numattr);
5334         if (certfile == NULL)
5335                 return (KMF_ERR_BAD_CRLFILE);

5337         dirpath = kmf_get_attr_ptr(KMF_DIRPATH_ATTR, attrlist, numattr);

5339         crlfile = get_fullpath(dirpath, crlfilename);

5341         if (crlfile == NULL)
5342                 return (KMF_ERR_BAD_CRLFILE);

5344         if (isdir(crlfile)) {
5345                 ret = KMF_ERR_BAD_CRLFILE;
5346                 goto end;
5347         }

5349         ret = kmf_is_crl_file(handle, crlfile, &format);
5350         if (ret != KMF_OK)
5351                 goto end;

5353         /* Read the CRL file and load it into a X509_CRL structure */
5354         in = BIO_new_file(crlfilename, "rb");
5355         if (in == NULL) {
5356                 SET_ERROR(kmfh, ERR_get_error());
5357                 ret = KMF_ERR_OPEN_FILE;
5358                 goto end;
5359         }

5361         if (format == KMF_FORMAT_ASN1) {
5362                 xcrl = d2i_X509_CRL_bio(in, NULL);
5363         } else if (format == KMF_FORMAT_PEM) {
5364                 xcrl = PEM_read_bio_X509_CRL(in, NULL, NULL, NULL);
5365         }

5367         if (xcrl == NULL) {
5368                 SET_ERROR(kmfh, ERR_get_error());
5369                 ret = KMF_ERR_BAD_CRLFILE;
5370                 goto end;
5371         }
5372         (void) BIO_free(in);

5374         /* Read the Certificate file and load it into a X509 structure */
5375         ret = kmf_is_cert_file(handle, certfile, &format);
5376         if (ret != KMF_OK)
5377                 goto end;

5379         in = BIO_new_file(certfile, "rb");
5380         if (in == NULL) {
5381                 SET_ERROR(kmfh, ERR_get_error());
5382                 ret = KMF_ERR_OPEN_FILE;
5383                 goto end;
5384         }
```

```
5386         if (format == KMF_FORMAT_ASN1) {
5387                 xcert = d2i_X509_bio(in, NULL);
5388         } else if (format == KMF_FORMAT_PEM) {
5389                 xcert = PEM_read_bio_X509(in, NULL, NULL, NULL);
5390         }

5392         if (xcert == NULL) {
5393                 SET_ERROR(kmfh, ERR_get_error());
5394                 ret = KMF_ERR_BAD_CERTFILE;
5395                 goto end;
5396         }

5398         /* Check if the certificate and the CRL have same issuer */
5399         if (X509_NAME_cmp(xcert->cert_info->issuer, xcrl->crl->issuer) != 0) {
5400                 ret = KMF_ERR_ISSUER;
5401                 goto end;
5402         }

5404         /* Check to see if the certificate serial number is revoked */
5405         revoke_stack = X509_CRL_get_REVOKED(xcrl);
5406         if (sk_X509_REVOKED_num(revoke_stack) <= 0) {
5407                 /* No revoked certificates in the CRL file */
5408                 SET_ERROR(kmfh, ERR_get_error());
5409                 ret = KMF_ERR_EMPTY_CRL;
5410                 goto end;
5411         }

5413         for (i = 0; i < sk_X509_REVOKED_num(revoke_stack); i++) {
5427                 /* LINTED E_BAD_PTR_CAST_ALIGN */
5414                 revoke = sk_X509_REVOKED_value(revoke_stack, i);
5415                 if (ASN1_INTEGER_cmp(xcert->cert_info->serialNumber,
5416                     revoke->serialNumber) == 0) {
5417                         break;
5418                 }
5419         }

5421         if (i < sk_X509_REVOKED_num(revoke_stack)) {
5422                 ret = KMF_OK;
5423         } else {
5424                 ret = KMF_ERR_NOT_REVOKED;
5425         }

5427 end:
5428         if (in != NULL)
5429                 (void) BIO_free(in);
5430         if (xcrl != NULL)
5431                 X509_CRL_free(xcrl);
5432         if (xcert != NULL)
5433                 X509_free(xcert);

5435         return (ret);
5436 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    2541 Fri May 16 10:29:54 2014
new/usr/src/lib/libpkg/Makefile.com
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#

#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

LIBRARY= libpkg.a
VERS=   .1

# include library definitions
OBJECTS=        \
                canonize.o   ckparam.o    ckvolseq.o \
                devtype.o    dstream.o    gpkglist.o \
                gpkgmap.o    isdir.o      logerr.o \
                mappath.o    ncgrpw.o     nhash.o \
                pkgexecl.o   pkgexecv.o   pkgmount.o \
                pkgtrans.o   ppkgmap.o \
                progerr.o    putcfile.o   rrmdir.o \
                runcmd.o     srchcfile.o  tputcfent.o \
                verify.o     security.o   pkgweb.o \
                pkgerr.o     keystore.o   p12lib.o \
                vfpops.o     fmkdir.o     pkgstr.o \
                handlelocalfs.o pkgserv.o


# include library definitions
include $(SRC)/lib/Makefile.lib

SRCDIR=         ../common

POFILE =        libpkg.po
MSGFILES =      $(OBJECTS:%.o=../common/%.i)
CLEANFILES +=   $(MSGFILES)

# This library is NOT lint clean

# openssl forces us to ignore dubious pointer casts, thanks to its clever
# use of macros for stack management.
LINTFLAGS=      -umx -errtags \
```



```
                -erroff=E_BAD_PTR_CAST_ALIGN,E_BAD_PTR_CAST
$(LINTLIB):=    SRCS = $(SRCDIR)/$(LINTSRC)


LIBS = $(DYNLIB) $(LINTLIB)


LDLIBS +=       -lc -lwanboot -lscf -ladm

# libcrypto and libssl have no lint library, and so can only be used when
# building
$(DYNLIB) := LDLIBS += -lcrypto -lssl
LDLIBS +=       -lc -lssl -lwanboot -lcrypto -lscf -ladm

CFLAGS +=       $(CCVERBOSE)
CERRWARN +=     -_gcc=-Wno-unused-label
CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-uninitialized
CERRWARN +=     -_gcc=-Wno-clobbered
CERRWARN +=     -_gcc=-Wno-switch
CERRWARN +=     -_gcc=-Wno-unused-value
CPPFLAGS +=     -I$(SRCDIR) -D_FILE_OFFSET_BITS=64

.KEEP_STATE:

all:    $(LIBS)

$(POFILE): $(MSGFILES)
        $(BUILDPO.msgfiles)

_msg: $(MSGDOMAINPOFILE)

lint: lintcheck

# include library targets
include $(SRC)/lib/Makefile.targ
include $(SRC)/Makefile.msg.targ
```

```
**********************************************************
    6803 Fri May 16 10:29:55 2014
new/usr/src/lib/libpkg/common/security.c
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
______unchanged_portion_omitted_

  79 /*
  80  * get_cert_chain - Builds a chain of certificates, from a given
  81  * user certificate to a trusted certificate.
  82  *
  83  * Arguments:
  84  * err - Error object to add errors to
  85  * cert - User cert to start with
  86  * cas - Trusted certs to use as trust anchors
  87  * chain - The resulting chain of certs (in the form of an
  88  * ordered set) is placed here.
  89  *
  90  * Returns:
  91  *   0 - Success - chain is stored in 'chain'.
  92  * non-zero - Failure, errors recorded in err
  93  */
  94 int
  95 get_cert_chain(PKG_ERR *err, X509 *cert, STACK_OF(X509) *clcerts,
  96     STACK_OF(X509) *cas, STACK_OF(X509) **chain)
  97 {
  98         X509_STORE_CTX  *store_ctx = NULL;
  99         X509_STORE      *ca_store = NULL;
 100         X509            *ca_cert = NULL;
 101         int i;
 102         int ret = 0;

 104         if ((ca_store = X509_STORE_new()) == NULL) {
 105                 pkgerr_add(err, PKGERR_NOMEM,
 106                     gettext(ERR_MEM));
 107                 ret = 1;
 108                 goto cleanup;
 109         }

 111         /* add all ca certs into the store */
 112         for (i = 0; i < sk_X509_num(cas); i++) {
 113                 /* LINTED pointer cast may result in improper alignment */
 113                 ca_cert = sk_X509_value(cas, i);
 114                 if (X509_STORE_add_cert(ca_store, ca_cert) == 0) {
 115                         pkgerr_add(err, PKGERR_NOMEM, gettext(ERR_MEM));
 116                         ret = 1;
 117                         goto cleanup;
 118                 }
 119         }

 121         /* initialize context object used during the chain resolution */

 123         if ((store_ctx = X509_STORE_CTX_new()) == NULL) {
 124                 pkgerr_add(err, PKGERR_NOMEM, gettext(ERR_MEM));
 125                 ret = 1;
 126                 goto cleanup;
 127         }

 129         (void) X509_STORE_CTX_init(store_ctx, ca_store, cert, clcerts);
 130         /* attempt to verify the cert, which builds the cert chain */
 131         if (X509_verify_cert(store_ctx) <= 0) {
 132                 pkgerr_add(err, PKGERR_CHAIN,
 133                     gettext(ERR_CERTCHAIN),
 134                     get_subject_display_name(cert),
```



```
 135                     X509_verify_cert_error_string(store_ctx->error));
 136                 ret = 1;
 137                 goto cleanup;
 138         }
 139         *chain = X509_STORE_CTX_get1_chain(store_ctx);

 141 cleanup:
 142         if (ca_store != NULL)
 143                 (void) X509_STORE_free(ca_store);
 144         if (store_ctx != NULL) {
 145                 (void) X509_STORE_CTX_cleanup(store_ctx);
 146                 (void) X509_STORE_CTX_free(store_ctx);
 147         }

 149         return (ret);
 150 }

 152 /*
 153  * Name:                get_subject_name
 154  * Description: Retrieves a name used for identifying a certificate's subject.
 155  *
 156  * Arguments:   cert - The certificate to get the name from
 157  *
 158  * Returns :    A static buffer containing the common name (CN) of the
 159  *              subject of the cert.
 160  *
 161  *              if the CN is not available, returns a string with the entire
 162  * X509 distinguished name.
 163  */
 164 char
 165 *get_subject_display_name(X509 *cert)
 166 {

 168         X509_NAME       *xname;
 169         static char     sname[ATTR_MAX];

 171         xname = X509_get_subject_name(cert);
 172         if (X509_NAME_get_text_by_NID(xname,
 173             NID_commonName, sname,
 174             ATTR_MAX) <= 0) {
 175                 (void) strncpy(sname,
 176                     X509_NAME_oneline(xname, NULL, 0), ATTR_MAX);
 177                     X509_NAME_oneline(xname,
 178                         NULL, 0), ATTR_MAX);
 177                 sname[ATTR_MAX - 1] = '\0';
 178         }
 179         return (sname);
 180 }

 182 /*
 183  * Name:                get_display_name
 184  * Description: Retrieves a name used for identifying a certificate's issuer.
 185  *
 186  * Arguments:   cert - The certificate to get the name from
 187  *
 188  * Returns :    A static buffer containing the common name (CN)
 189  *              of the issuer of the cert.
 190  *
 191  *              if the CN is not available, returns a string with the entire
 192  *              X509 distinguished name.
 193  */
 194 char
 195 *get_issuer_display_name(X509 *cert)
 196 {

 198         X509_NAME       *xname;
```

```
199         static char     sname[ATTR_MAX];

201         xname = X509_get_issuer_name(cert);
202         if (X509_NAME_get_text_by_NID(xname,
203             NID_commonName, sname,
204             ATTR_MAX) <= 0) {
205                 (void) strncpy(sname,
206                     X509_NAME_oneline(xname, NULL, 0), ATTR_MAX);
208                     X509_NAME_oneline(xname,
209                         NULL, 0), ATTR_MAX);
207                 sname[ATTR_MAX - 1] = '\0';
208         }
209         return (sname);
210 }
```

*unchanged_portion_omitted_*

```
**********************************************************
    2761 Fri May 16 10:29:55 2014
new/usr/src/lib/libwanboot/Makefile.com
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# Copyright (c) 2012 by Delphix. All rights reserved.
#

LIBRARY =       libwanboot.a
VERS =          .1

# List of locally located modules.
LOC_DIR =       ../common
LOC_OBJS =      socket_inet.o bootinfo_aux.o
LOC_SRCS =      $(LOC_OBJS:%.o=$(LOC_DIR)/%.c)

# List of common wanboot objects.
COM_DIR =       ../../../common/net/wanboot
COM_OBJS =      auxutil.o \
                boot_http.o \
                bootconf.o \
                bootconf_errmsg.o \
                bootinfo.o \
                bootlog.o \
                http_errorstr.o \
                p12access.o \
                p12auxpars.o \
                p12auxutl.o \
                p12err.o \
                p12misc.o \
                parseURL.o
COM_SRCS =      $(COM_OBJS:%.o=$(COM_DIR)/%.c)

# List of common DHCP modules.
DHCP_DIR =      $(SRC)/common/net/dhcp
DHCP_OBJS =     dhcpinfo.o
DHCP_SRCS =     $(DHCP_OBJS:%.o=$(DHCP_DIR)/%.c)

OBJECTS =       $(LOC_OBJS) $(COM_OBJS) $(DHCP_OBJS)

include ../../Makefile.lib
```



```
LIBS +=         $(LINTLIB)
LDLIBS +=       -lnvpair -lresolv -lnsl -lsocket -ldevinfo -ldhcputil \
                -linetutil -lc

# libcrypto and libssl have no lint library, so we can only use it when
# building
$(DYNLIB) := LDLIBS += -lcrypto -lssl

                -linetutil -lc -lcrypto -lssl
CPPFLAGS =      -I$(SRC)/common/net/wanboot/crypt $(CPPFLAGS.master)
CERRWARN +=     -_gcc=-Wno-switch
CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-unused-value
CERRWARN +=     -_gcc=-Wno-uninitialized

# Must override SRCS from Makefile.lib since sources have
# multiple source directories.
SRCS =          $(LOC_SRCS) $(COM_SRCS) $(DHCP_SRCS)

# Must define location of lint library source.
SRCDIR =        $(LOC_DIR)
$(LINTLIB) :=   SRCS = $(SRCDIR)/$(LINTSRC)

# OpenSSL requires us to turn this off
LINTFLAGS   +=    -erroff=E_BAD_PTR_CAST_ALIGN
LINTFLAGS64 +=    -erroff=E_BAD_PTR_CAST_ALIGN

CFLAGS +=       $(CCVERBOSE)
CPPFLAGS +=     -I$(LOC_DIR) -I$(COM_DIR) -I$(DHCP_DIR)

.KEEP_STATE:

all: $(LIBS)

lint: lintcheck

pics/%.o: $(COM_DIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

pics/%.o: $(DHCP_DIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

include ../../Makefile.targ
```

```
**********************************************************
    2438 Fri May 16 10:29:55 2014
new/usr/src/lib/pkcs11/pkcs11_tpm/Makefile.com
4853 illumos-gate is not lint-clean when built with openssl 1.0
Reviewed by Keith Wesolowski <keith.wesolowski@joyent.com>
Reviewed by Alexander Eremin <alexander.eremin@nexenta.com>
**********************************************************
```

```
#
# CDDL HEADER START
#
# The contents of this file are subject to the terms of the
# Common Development and Distribution License (the "License").
# You may not use this file except in compliance with the License.
#
# You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
# or http://www.opensolaris.org/os/licensing.
# See the License for the specific language governing permissions
# and limitations under the License.
#
# When distributing Covered Code, include this CDDL HEADER in each
# file and include the License file at usr/src/OPENSOLARIS.LICENSE.
# If applicable, add the following below this CDDL HEADER, with the
# fields enclosed by brackets "[]" replaced with your own identifying
# information: Portions Copyright [yyyy] [name of copyright owner]
#
# CDDL HEADER END
#
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
LIBRARY =       pkcs11_tpm.a
VERS =          .1

OBJECTS= api_interface.o \
        apiutil.o \
        asn1.o \
        cert.o \
        data_obj.o \
        decr_mgr.o \
        dig_mgr.o \
        encr_mgr.o \
        globals.o \
        hwf_obj.o \
        key.o \
        key_mgr.o \
        loadsave.o \
        log.o \
        mech_md5.o \
        mech_rsa.o \
        mech_sha.o \
        new_host.o \
        obj_mgr.o \
        object.o \
        sess_mgr.o \
        sign_mgr.o \
        template.o \
        tpm_specific.o \
        utility.o \
        verify_mgr.o


include $(SRC)/lib/Makefile.lib

SRCDIR= ../common

SRCS=   $(OBJECTS:%.o=$(SRCDIR)/%.c)
```



```
#       set signing mode
POST_PROCESS_SO +=      ; $(ELFSIGN_CRYPTO)

ROOTLIBDIR=$(ROOT)/usr/lib/security
ROOTLIBDIR64=$(ROOT)/usr/lib/security/$(MACH64)

LIBS=$(DYNLIB) $(DYNLIB64)

TSSROOT=$(ADJUNCT_PROTO)
TSPILIBDIR=$(TSSROOT)/usr/lib
TSPIINCDIR=$(TSSROOT)/usr/include
TSSLIB=-L$(TSPILIBDIR)
TSSLIB64=-L$(TSPILIBDIR)/$(MACH64)
TSSINC=-I$(TSPIINCDIR)

LDLIBS += $(TSSLIB) -L$(ADJUNCT_PROTO)/lib -lc -luuid -lmd -ltspi

# libcrypto has no lint library, so we can only use it when
# building
$(LIBS) := LDLIBS += -lcrypto

LDLIBS += $(TSSLIB) -L$(ADJUNCT_PROTO)/lib -lc -luuid -lmd -ltspi -lcrypto
CPPFLAGS += -xCC -D_POSIX_PTHREAD_SEMANTICS $(TSSINC)
CPPFLAGS64 += $(CPPFLAGS)
C99MODE=        $(C99_ENABLE)

CERRWARN +=     -_gcc=-Wno-parentheses
CERRWARN +=     -_gcc=-Wno-unused-label
CERRWARN +=     -_gcc=-Wno-uninitialized

LINTSRC= $(OBJECTS:%.o=$(SRCDIR)/%.c)

$(LINTLIB):=    SRCS    =       $(SRCDIR)/$(LINTSRC)
LINTSRC= $(SRCS)

CLOBBERFILES += C.ln

.KEEP_STATE:

all: $(LIBS)

lint: $$(LINTSRC)
        $(LINT.c) $(LINTCHECKFLAGS) $(LINTSRC) $(LDLIBS)

pics/%.o: $(SRCDIR)/%.c
        $(COMPILE.c) -o $@ $<
        $(POST_PROCESS_O)

include $(SRC)/lib/Makefile.targ
```