ภาคผนวก ข

ภาคผนวก ข.1

แสดง Source Code ใน /etc/krb5.conf

```
[libdefaults]
        proxiable = true
        forwardable = true
        default_realm = SOMEWHERE.COM
        noaddresses = false
        extra_address = 203.144.155.1

[realms]
        SOMEWHERE.COM = {
                kdc = 203.144.155.2:88
                default_domain = SOMEWHERE.COM
        }

[domain_realm]
        .somewhere.com = SOMEWHERE.COM
        somewhere.com = SOMEWHERE.COM

[logging]
        kdc = CONSOLE
```

ภาคผนวก ข.2

แสดง Source Code K5_Kinit ใน /kinit.c

```
static int
k5_kinit(opts, k5)
        struct k_opts* opts;
        struct k5_data* k5;

{
        char* progname = progname_v5;
        int notix =1;
        krb5_keytab keytab = 0;
        krb5_creds my_creds;
        krb5_error_code code = 0;
        krb5_get_init_creds_opt options;
        krb5_address **addresses = NULL;

        if (!got_k5)
                return 0;

        krb5_get_init_creds_opt_init(&options);
        memset(&my_creds, 0, sizeof (my_creds));

/*
        From this point on, we can goto cleanup because my cred is initialized

*/
```

```
if (opts -> lifetime)
        krb5_get_init_creds_opt_set_tkt_life(&options, opts -> lifetime);
if (opts -> rlife)
        krb5_get_init_creds_opt_set_renew_life(&options, opts -> rlife);
if (opts -> forwardable)
        krb5_get_init_creds_opt_set_forwardable(&options, 1);
if (opts -> not_forwardable)
        krb5_get_init_creds_opt_set_forwardable(&options, 0);
if (opts -> proxiable)
        krb5_get_init_creds_opt_set_proxiable(&options, 1);
if (opts -> not_proxiable)
        krb5_get_init_creds_opt_set_proxiable(&options, 0);
if (opts -> addresses)

{
        code = krb5_os_localaddr(k5 -> ctx, &addresses);
        if (code !=0)    {
                com_err(progname, code, "getting local addresses");
                goto cleanup;
        }
        krb5_get_init_creds_opt_set_address_list(&options, addresses);
}

if (opts -> no_addresses)
        krb5_get_init_creds_opt_set_addresss_list(&options, NULL);

if ((opts -> action == INIT_KT) && opts -> keytab_name)

{
```

```
        code = krb5_kt_resolve(k5 -> ctx, opts -> keytab_name, &keytab);
        if (code != 0)   {
                com_err(progname, code, "resolving keytab %s",
                        opts -> keytab_name);
                goto cleanup;

        }
}
```

ภาคผนวก ข.3

แสดง Source Code main() ใน /kinit.c

```
int
main (argc, argv)
        int argc;
        char **argv;

{
        struct k_opts opts;
        struct k5_data k5;
        struct k4_data k4;
        char *progname;

        progname = GET_PROGNAME(argv[0]);
        progname_v5 = getvprogname("5", progname);

# ifdef  KRB5_KRB4_COMPAT

        progname_v4 = getvprogname ("4", progname);
        progname_v524 = getvprogname("524", progname);

#endif

/* Ensure we can be driven from a pipe */
        if (!isatty(fileno(stdin)))
                setvbuf (stdin, 0, _IONBF, 0);
```

```
        if (!isatty(fileno(stdout)))
                setvbuf (stdout, 0, _IONBF, 0);
        if (!isatty(fileno(stderr)))
                setvbuf (stderr, 0, _IONBF, 0);

/*
        This is where we would put in code to dynamically load kerberos libraries.
        Currently, we just get them implicitly.
*/
        got_k5 = 1;

# ifdef  KRB5_KRB4_COMPAT

        got_k4 = 1;

# endif

        memset(&opts, 0, sizeof (opts));
        opts.action = INIT_PW;

        memset(&k5, 0, sizeof(k5));
        memset(&k4, 0, sizeof(k4));

        parse_options(argc, argv, &opts, progname);

        got_k5 = k5_begin(&opts, &k5, &k4);
        got_k4 = k4_begin(&opts, &k4);

        authed_k5 = k5_kinit(&opts, &k5);
```

```
# ifdef HAVE_KRB524

        if (authed_k5)
                authed_k4 = try_convert524(&k5);

#endif

        if (!authed_k4)
                authed_k4 = k4_kinit(&opts, &k4, K5.ctx);

# ifdef  KRB5_KRB4_COMPAT

                memset(stash_password, 0, sizeof(stash_password));

# endif

        if (authed_k5 && opts.verbose)
                fprintf(stderr, "Authenicated to Kerberos v5\n");
        if (authed_k4 && opts.verbose)
                fprintf(stderr, "Authenticated to Kerberos v4\n");

        k5_end(&k5);
        k4_end(&k4);

        if ((got_k5 && !authed_k5) || (got_k4 && !authed_k4) ||
                (!got_k5 && !got_k4))
                exit (1);
        return 0;
}
```