# KADM5 Admin API
# Unit Test Description*

### Jonathan I. Kamens

### January 6, 2007

## 1   Introduction

The following is a description of a black-box unit test of the KADM5 API. Each API function is listed, followed by the tests that shoud be performed on it.

The tests described here are based on the "Kerberos Administration System KADM5 API Functional Specifications", revision 1.68. This document was originally written based on the OpenVision API functional specifications, version 1.41, dated August 18, 1994, and many indications of the original version remain.

All tests which test for success should verify, using some means other than the return value of the function being tested, that the requested operation was successfully performed. For example: for init, test that other operations can be performed after init; for destroy, test that other operations can't be performed after destroy; for modify functions, verify that all modifications to the database which should have taken place did, and that the new, modified data is in effect; for get operations, verify that the data retrieved is the data that should actually be in the database.

The tests would be better if they compared the actual contents of the database before and after each test, rather than relying on the KADM5 API to report the results of changes.

Similarly, all tests which test for failure should verify that the no component of the requested operation took place. For example: if init fails, other operations should not work. If a modify fails, all data in the database should be the same as it was before the attempt to modify, and the old data should still be what is enforced. Furthermore, tests which test for failure should verify that the failure code returned is correct for the specific failure condition tested.

Most of the tests listed below should be run twice – once locally on the server after linking against the server API library, and once talking to the server via authenticated Sun RPC after linking against the client API library. Tests which should only be run locally or via RPC are labelled with a "local" or "RPC".

Furthermore, in addition to the tests labelled below, a test should be implemented to verify that a client can't perform operations on the server through the client API library when it's linked against standard Sun RPC instead of OpenV*Secure's authenticated Sun RPC. This will require a client with a modified version of ovsec_kadm_init which doesn't call auth_gssapi_create. This client should call this modified ovsec_kadm_init and then call some other admin API function, specifying arguments to both functions that would work if the authenticated Sun RPC had been used, but shouldn't if authentication wasn't used. The test should verify that the API function call after the init doesn't succeed.

There is also another test to see if all the API functions handle getting an invalid server handle correctly. This is not done as part of the tests that are run through the TCL program cause the TCL program has no way of invalidating a server handle. So there is a program that calls init and changes the handle magic number, and then attempts to call each API function with the corrupted server handle.

A number of tests have been added or changed to correspond with KADM5 API version 2. Tests which are only performed against the newer version specify the version number in the test description.

---

*api-unit-test.tex 17360 2005-08-25 23:41:34Z raeburn

# 2   ovsec_kadm_init

**Number:** 1

**Reason:** An empty string realm is rejected.

**Status:** Implemented

**V2 note:** The empty string is now passed as the realm field of the parameters structure.

**Number:** 2

**Reason:** A realm containing invalid characters is rejected.

**Status:** Implemented

**V2 note:** The invalid character is now passed as the realm field of the parameters structure.

**Number:** 2.5

**Reason:** A non-existent realm is rejected.

**Status:** Implemented

**V2 note:** The non-existent realm is now passed as the realm field of the parameters structure.

**Number:** 3

**Reason:** A bad service name representing an existing principal (different from the client principal) is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 4

**Reason:** A bad service name representing a non-existent principal is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 5

**Reason:** A bad service name identical to the (existing) client name is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 6

**Reason:** A null password causes password prompting.

**Status:** Implemented

**Number:** 7

**Reason:** An empty-string causes password prompting

**Status:** Implemented

**Number:** 8

**Reason:** An incorrect password which is the password of another user is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 9

**Reason:** An incorrect password which isn't the password of any user is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 10

**Reason:** A null client_name is rejected.

**Status:** Implemented

**Number:** 12

**Reason:** A client_name referring to a non-existent principal in the default realm is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 13

**Reason:** A client_name referring to a non-existent principal with the local realm specified explicitly is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 14

**Reason:** A client_name referring to a non-existent principal in a nonexistent realm is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 15

**Reason:** A client_name referring to an existing principal in a nonexistent realm is rejected.

**Conditions:** RPC

**Status:** Implemented

**Number:** 16

**Reason:** Valid invocation.

**Status:** Implemented

**Number:** 17

**Reason:** Valid invocation (explicit client realm).

**Status:** Implemented

**Number:** 18

**Reason:** Valid invocation (CHANGEPW_SERVICE).

**Status:** Implemented

**Number:** 19

**Reason:** Valid invocation (explicit service realm).

**Status:** Implemented

**V2 note:** The explicit realm is now passed as the realm field of the configuration parameters.

**Number:** 20

**Reason:** Valid invocation (database access allowed after init).

**Status:** Implemented

**Number:** 22

**Reason:** A null password causes master-key prompting.

**Conditions:** local

**Status:** Implemented

**V2 note:** Obsolete.

**Number:** 22.5

**Reason:** A empty string password causes master-key prompting.

**Conditions:** local

**Status:** Implemented

**V2 note:** Obsolete.

**Number:** 24

**Reason:** Null service name is ignored in local invocation.

**Conditions:** local

**Status:** Implemented

**Number:** 25

**Reason:** Non-null service name is ignored in local invocation.

**Conditions:** local

**Status:** Implemented

**Number:** 30

**Reason:** Can init after failed init attempt.

**Conditions:** local

**Status:** Implemented

**Number:** 31

**Priority:** High

**Reason:** Return BAD_STRUCT_VERSION when the mask bits are set to invalid values

**Status:** Implemented

**Number:** 32

**Priority:** High

**Reason:** Return BAD_STRUCT_VERSION when the mask bits are not set

**Status:** Implemented

**Number:** 33

**Priority:** High

**Reason:** Return OLD_STRUCT_VERSION when attempting to use an old/unsupported structure version

**Status:** Implemented

**Number:** 34

**Priority:** High

**Reason:** Return NEW_STRUCT_VERSION when attempting to use a newer version of of the structure then what is supported

**Status:** Implemented

**Number:** 35

**Priority:** High

**Reason:** Return BAD_API_VERSION when the mask bits are set to invalid values

**Status:** Implemented

**Number:** 36

**Priority:** High

**Reason:** Return BAD_API_VERSION when the mask bits are not set

**Status:** Implemented

**Number:** 37

**Priority:** High

**Reason:** Return OLD_LIB_API_VERSION when using an old/unsupported api version number

**Conditions:** RPC

**Status:** Implemented

**Number:** 38

**Priority:** High

**Reason:** Return OLD_SERVER_API_VERSION attempting to use an old/unsupported api version number

**Conditions:** local

**Status:** Implemented

**Number:** 39

**Priority:** High

**Reason:** Return NEW_LIB_API_VERSION when using a newer api version number then supported

**Conditions:** RPC

**Status:** Implemented

**Number:** 40

**Priority:** High

**Reason:** Return NEW_SERVER_API_VERSION when using a newer api version number then supported

**Conditions:** local

**Status:** Implemented

**Number:** 41

**Priority:** High

**Reason:** Return BAD_XXX_VERSION when the API and the structure version numbers are reversed

**Status:** Implemented

**Number:** 42

**Priority:** High

**Reason:** Succeeds when using valid api and struct version numbers and masks

**Status:** Implemented

**Number:** 43

**Priority:** Low

**Reason:** Returns two different server handle when called twice with same info

**Number:** 44

**Priority:** Low

**Reason:** Returns two different server handles when called twice with different info

**Number:** 45

**Priority:** Bug fix, secure-install/3390

**Reason:** Returns SECURE_PRINC_MISSING when ADMIN_SERVICE does not exist.

**Status:** Implemented

**Number:** 46

**Priority:** Bug fix, secure-install/3390

**Reason:** Returns SECURE_PRINC_MISSING when CHANGEPW_SERVICE does not exist.

**Status:** Implemented

**Number:** 100

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the profile field of the configuration parameters, if set.

**Status:** Implemented

**Number:** 101

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the kadmind_port field of the configuration parameters, if set.

**Conditions:** RPC

**Status:** Implemented

**Number:** 102

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the admin_server field of the configuration parameters, if set with only an admin server name.

**Conditions:** RPC

**Status:** Implemented

**Number:** 102.5

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the admin_server field of the configuratin parameters, if set with a host name and port number.

**Conditions:** RPC

**Number:** 103

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the dbname field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 104

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the admin_dbname field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 105

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the admin_lockfile field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 106

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the mkey_from_kbd field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 107

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the stash_file field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 108

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the mkey_name field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 109

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the max_life field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 110

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the max_rlife field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 111

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the expiration field of the configuration parameters, if set.

**Status:** Implemented

**Conditions:** local

**Number:** 112

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the flags field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 113

**Version:** KADM5_API_VERSION_2

**Reason:** Obeys the keysalts and num_keysalts field of the configuration parameters, if set.

**Conditions:** local

**Status:** Implemented

**Number:** 114

**Version:** KADM5_API_VERSION_2

**Reason:** Returns KADM5_BAD_SERVER_PARAMS if any client-only parameters are specified to server-side init.

**Conditions:** local

**Status:** Implemented

**Number:** 115

**Version:** KADM5_API_VERSION_2

**Reason:** Returns KADM5_BAD_CLIENT_PARAMS if any client-only parameters are specified to server-side init.

**Conditions:** RPC

**Status:** Implemented

**Number:** 116

**Version:** KADM5_API_VERSION_2

**Reason:** Two calls to init with clients having different privileges succeedes, and both clients maintain their correct privileges.

**Priority:** Bug fix

**Conditions:** RPC

**Status:** Implemented

**Number:** 117

**Version:** KADM5_API_VERSION_2

**Reason:** The max_life field defaults to value specified in the API Functional Specification when kdc.conf is unreadable.

**Priority:** Bug fix, krb5-admin/18

**Conditions:** local

**Status:** Implemented

**Number:** 150

**Version:** KADM5_API_VERSION_2

**Reason:** init_with_creds works when given an open ccache with a valid credential for ADMIN_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 151

**Version:** KADM5_API_VERSION_2

**Reason:** init_with_creds works when given an open ccache with a valid credential for CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 152

**Version:** KADM5_API_VERSION_2

**Reason:** init_with_creds fails with KRB5_FCC_NOFILE (was KADM5_GSS_ERROR) when given an open ccache with no credentials.

**Conditions:** RPC

**Status:** Implemented

**Number:** 153

**Version:** KADM5_API_VERSION_2

**Reason:** init_with_creds fails with KRB5_CC_NOTFOUND (was KADM5_GSS_ERROR) when given an open ccache without credentials for ADMIN_SERVICE or CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 154

**Version:** KADM5_API_VERSION_2

**Reason:** If the KRB5_KDC_PROFILE environment variable is set to a filename that does not exist, init fails with ENOENT.

**Conditions:** RPC

**Status:** Implemented

# 3   ovsec_kadm_destroy

**Number:** 1
**Reason:** Valid invocation.
**Status:** Implemented

**Number:** 8
**Reason:** Database can be reinitialized after destroy.
**Status:** Implemented

**Number:** 9
**Priority:** High
**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in
**Status:** Implemented

**Number:** 10
**Priority:** Low
**Reason:** Connects to correct server when mutliple handles exist
**Conditions:** client

# 4   ovsec_kadm_create_principal

**Number:** 2
**Reason:** Fails on null princ argument.
**Status:** Implemented

**Number:** 3
**Reason:** Fails on null password argument.
**Status:** Implemented

**Number:** 4
**Reason:** Fails on empty-string password argument.
**Status:** Implemented

**Number:** 5
**Reason:** Fails when mask contains undefined bit.
**Status:** Implemented

**Number:** 6
**Reason:** Fails when mask contains LAST_PWD_CHANGE bit.
**Status:** Implemented

**Number:** 7
**Reason:** Fails when mask contains MOD_TIME bit.
**Status:** Implemented

**Number:** 8

**Reason:** Fails when mask contains MOD_NAME bit.

**Status:** Implemented

**Number:** 9

**Reason:** Fails when mask contains MKVNO bit.

**Status:** Implemented

**Number:** 10

**Reason:** Fails when mask contains AUX_ATTRIBUTES bit.

**Status:** Implemented

**Number:** 11

**Reason:** Fails when mask contains POLICY_CLR bit.

**Status:** Implemented

**Number:** 12

**Reason:** Fails for caller with no access bits.

**Status:** Implemented

**Number:** 13

**Reason:** Fails when caller has "get" access and not "add".

**Conditions:** RPC

**Status:** Implemented

**Number:** 14

**Reason:** Fails when caller has "modify" access and not "add".

**Conditions:** RPC

**Status:** Implemented

**Number:** 15

**Reason:** Fails when caller has "delete" access and not "add".

**Conditions:** RPC

**Status:** Implemented

**Number:** 16

**Reason:** Fails when caller connected with CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 17

**Reason:** Fails on attempt to create existing principal.

**Status:** Implemented

**Number:** 18

**Reason:** Fails when password is too short.
**Status:** Implemented

**Number:** 19
**Reason:** Fails when password has too few classes.
**Status:** Implemented

**Number:** 20
**Reason:** Fails when password is in dictionary.
**Status:** Implemented

**Number:** 21
**Reason:** Nonexistent policy is rejected.
**Status:** Implemented

**Number:** 22
**Reason:** Fails on invalid principal name.
**Status:** Implemented

**Number:** 23
**Reason:** Valid invocation.
**Status:** Implemented

**Number:** 24
**Reason:** Succeeds when caller has "add" access and another one.
**Status:** Implemented

**Number:** 28
**Reason:** Succeeds when assigning policy.
**Status:** Implemented

**Number:** 29
**Priority:** High
**Reason:** Allows 0 (never) for princ_expire_time.
**Status:** Implemented

**Number:** 30
**Reason:** Allows 0 (never) for pw_expiration when there's no policy.
**Status:** Implemented

**Number:** 31
**Reason:** Allows 0 (never) for pw_expiration when there's a policy with 0 for pw_max_life.
**Status:** Implemented

**Number:** 32

**Reason:** Accepts 0 (never) for pw_expiration when there's a policy with non-zero pw_max_life, and sets pw_expiration to zero.

**Status:** Implemented

**Number:** 33

**Reason:** Accepts and sets non-zero pw_expiration when no policy.

**Status:** Implemented

**Number:** 34

**Reason:** Accepts and sets non-zero pw_expiration when there's a policy with zero pw_max_life.

**Status:** Implemented

**Number:** 35

**Reason:** Accepts and sets non-zero pw_expiration when there's a policy with pw_max_life later than the specified pw_expiration.

**Status:** Implemented

**Number:** 36

**Reason:** Accepts and sets non-zero pw_expiration greater than now_pw_max_life.

**Status:** Implemented

**Number:** 37

**Priority:** High

**Reason:** Sets pw_expiration to 0 (never) if there's no policy and no specified pw_expiration.

**Status:** Implemented

**Number:** 38

**Priority:** High

**Reason:** Sets pw_expiration to 0 (never) if it isn't specified and the policy has a 0 (never) pw_max_life.

**Status:** Implemented

**Number:** 39

**Priority:** High

**Reason:** Sets pw_expiration to now + pw_max_life if it isn't specified and the policy has a non-zero pw_max_life.

**Status:** Implemented

**Number:** 40

**Priority:** High

**Reason:** Allows 0 (forever) for max_life.

**Status:** Implemented

**Number:** 41

**Priority:** High

**Reason:** Doesn't modify or free mod_name on success.

**Number:** 42

**Priority:** High

**Reason:** Doesn't modify or free mod_name on failure.

**Number:** 43

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 44

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

# 5 ovsec_kadm_delete_principal

**Number:** 2

**Reason:** Fails on null principal.

**Status:** Implemented

**Number:** 5

**Priority:** High

**Reason:** Fails on nonexistent principal.

**Status:** Implemented

**Number:** 6

**Priority:** High

**Reason:** Fails when caller connected with CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 7

**Priority:** High

**Reason:** Fails if caller has "add" access and not "delete".

**Conditions:** RPC

**Status:** Implemented

**Number:** 8

**Priority:** High

**Reason:** Fails if caller has "modify" access and not "delete".

**Conditions:** RPC

**Status:** Implemented

**Number:** 9

**Priority:** High

**Reason:** Fails if caller has "get" access and not "delete".

**Conditions:** RPC

**Status:** Implemented

**Number:** 10

**Priority:** High

**Reason:** Fails if caller has no access bits.

**Conditions:** RPC

**Status:** Implemented

**Number:** 11

**Priority:** High

**Reason:** Valid invocation.

**Status:** Implemented

**Number:** 12

**Priority:** High

**Reason:** Valid invocation (on principal with policy).

**Status:** Implemented

**Number:** 13

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 14

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

# 6   ovsec_kadm_modify_principal

**Number:** 2

**Priority:** High

**Reason:** Fails if user connected with CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 3

**Reason:** Fails on mask with undefined bit set.

**Status:** Implemented

**Number:** 4

**Reason:** Fails on mask with PRINCIPAL set.

**Status:** Implemented

**Number:** 5
**Priority:** High
**Reason:** Fails on mask with LAST_PWD_CHANGE set.
**Status:** Implemented

**Number:** 6
**Reason:** Fails on mask with MOD_TIME set.
**Status:** Implemented

**Number:** 7
**Reason:** Fails on mask with MOD_NAME set.
**Status:** Implemented

**Number:** 8
**Reason:** Fails on mask with MKVNO set.
**Status:** Implemented

**Number:** 9
**Priority:** High
**Reason:** Fails on mask with AUX_ATTRIBUTES set.
**Status:** Implemented

**Number:** 10
**Reason:** Fails on nonexistent principal.
**Status:** Implemented

**Number:** 11
**Priority:** High
**Reason:** Fails for user with no access bits.
**Conditions:** RPC
**Status:** Implemented

**Number:** 12
**Priority:** High
**Reason:** Fails for user with "get" access.
**Conditions:** RPC
**Status:** Implemented

**Number:** 13
**Priority:** High
**Reason:** Fails for user with "add" access.
**Conditions:** RPC
**Status:** Implemented

**Number:** 14

**Priority:** High

**Reason:** Fails for user with "delete" access.

**Conditions:** RPC

**Status:** Implemented

**Number:** 15

**Priority:** High

**Reason:** Succeeds for user with "modify" access.

**Conditions:** RPC

**Status:** Implemented

**Number:** 16

**Reason:** Succeeds for user with "modify" and another access.

**Conditions:** RPC

**Status:** Implemented

**Number:** 17

**Priority:** High

**Reason:** Fails when nonexistent policy is specified.

**Status:** Implemented

**Number:** 18

**Priority:** High

**Reason:** Succeeds when existent policy is specified.

**Status:** Implemented

**Number:** 19

**Reason:** Updates policy count when setting policy from none.

**Status:** Implemented

**Number:** 20

**Reason:** Updates policy count when clearing policy from set.

**Status:** Implemented

**Number:** 21

**Reason:** Updates policy count when setting policy from other policy.

**Status:** Implemented

**Number:** 21.5

**Reason:** Policy reference count remains unchanged when policy is changed to itself.

**Status:** Implemented.

**Number:** 22

**Reason:** Allows 0 (never) for pw_expiration when there's no policy.

**Status:** Implemented

**Number:** 23

**Reason:** Allows 0 (never) for pw_expiration when there's a policy with 0 for pw_max_life.

**Status:** Implemented

**Number:** 24

**Reason:** Accepts 0 (never) for pw_expiration when there's a policy with non-zero pw_max_life, but actually sets pw_expiration to last_pwd_change + pw_max_life.

**Status:** Implemented

**Number:** 25

**Reason:** Accepts and sets non-zero pw_expiration when no policy.

**Status:** Implemented

**Number:** 26

**Reason:** Accepts and sets non-zero pw_expiration when there's a policy with zero pw_max_life.

**Status:** Implemented

**Number:** 27

**Reason:** Accepts and sets non-zero pw_expiration when there's a policy with pw_max_life later than the specified pw_expiration.

**Status:** Implemented

**Number:** 28

**Reason:** Accepts non-zero pw_expiration and limits it to last_pwd_change + pw_max_life when it's later than last_pwd_change + non-zero pw_max_life in policy.

**Status:** Implemented

**Number:** 29

**Priority:** High

**Reason:** Sets pw_expiration to 0 (never) when a policy is cleared and no pw_expiration is specified.

**Status:** Implemented

**Number:** 30

**Priority:** High

**Reason:** Sets pw_expiration to 0 (never) if it isn't specified and the new policy has a 0 (never) pw_max_life.

**Status:** Implemented

**Number:** 31

**Priority:** High

**Reason:** Sets pw_expiration to now + pw_max_life if it isn't specified and the new policy has a non-zero pw_max_life.

**Status:** Implemented

**Number:** 32

**Priority:** High

**Reason:** Accepts princ_expire_time change.

**Status:** Implemented

**Number:** 33

**Priority:** High

**Reason:** Accepts attributes change.

**Status:** Implemented

**Number:** 33.25

**Priority:** High

**Reason:** Accepts attributes change (KRB5_KDB_REQUIRES_PW_CHANGE).

**Status:** Implemented

**Number:** 33.5

**Priority:** High

**Reason:** Accepts attributes change (KRB5_DISALLOW_TGT_BASE).

**Status:** Implemented

**Number:** 33.75

**Priority:** High

**Reason:** Accepts attributes change (KRB5_PW_CHANGE_SERVICE).

**Status:** Implemented

**Number:** 34

**Priority:** High

**Reason:** Accepts max_life change.

**Status:** Implemented

**Number:** 35

**Priority:** High

**Reason:** Accepts kvno change.

**Status:** Implemented

**Number:** 36

**Reason:** Behaves correctly when policy is set to the same as it was before.

**Status:** Implemented

**Number:** 37

**Reason:** Behaves properly when POLICY_CLR is specified and there was no policy before.

**Status:** Implemented

**Number:** 38

**Priority:** High

**Reason:** Accepts 0 (never) for princ_expire_time.

**Status:** Implemented

**Number:** 39

**Priority:** High

**Reason:** Accepts 0 for max_life.

**Status:** Implemented

**Number:** 40

**Reason:** Rejects null principal argument.

**Status:** Implemented

**Number:** 41

**Priority:** High

**Reason:** Doesn't modify or free mod_name on success.

**Number:** 42

**Priority:** High

**Reason:** Doesn't modify or free mod_name on failure.

**Number:** 43

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 44

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

**Number:** 100

**Version:** KADM5_API_VERSION_2

**Priority:** bug-fix

**Reason:** Accepts max_rlife change.

**Status:** Implemented

**Number:** 101

**Version:** KADM5_API_VERSION_2

**Reason:** Rejects last_success change.

**Status:** Implemented

**Number:** 102

**Version:** KADM5_API_VERSION_2

**Reason:** Rejects last_failed change.

**Status:** Implemented

**Number:** 103

**Version:** KADM5_API_VERSION_2

**Reason:** Rejects fail_auth_count change.
**Status:** Implemented

**Number:** 103.5
**Version:** KADM5_API_VERSION_2
**Reason:** Rejects key_data change.
**Status:** Implemented

**Number:** 104
**Version:** KADM5_API_VERSION_2
**Reason:** Accepts tl_data change when all types are greater than 256.
**Status:** Implemented

**Number:** 105
**Version:** KADM5_API_VERSION_2
**Reason:** Returns KADM5_BAD_TL_TYPE when given tl_data with a type less than 256.
**Status:** Implemented

# 7   ovsec_kadm_rename_principal

**Number:** 2
**Priority:** High
**Reason:** Fails if user connected with CHANGEPW_SERVICE.
**Conditions:** RPC
**Status:** Implemented

**Number:** 3
**Priority:** High
**Reason:** Fails for user with no access bits.
**Conditions:** RPC
**Status:** Implemented

**Number:** 4
**Reason:** Fails for user with "modify" access and not "add" or "delete".
**Conditions:** RPC
**Status:** Implemented

**Number:** 5
**Reason:** Fails for user with "get" access and not "add" or "delete".
**Conditions:** RPC
**Status:** Implemented

**Number:** 6
**Reason:** Fails for user with "modify" and "add" but not "delete".

**Conditions:** RPC
**Status:** Implemented

**Number:** 7
**Reason:** Fails for user with "modify" and "delete" but not "add".
**Conditions:** RPC
**Status:** Implemented

**Number:** 8
**Reason:** Fails for user with "get" and "add" but not "delete".
**Conditions:** RPC
**Status:** Implemented

**Number:** 9
**Reason:** Fails for user with "get" and "delete" but not "add."
**Conditions:** RPC
**Status:** Implemented

**Number:** 10
**Reason:** Fails for user with "modify", "get" and "add", but not "delete".
**Conditions:** RPC
**Status:** Implemented

**Number:** 11
**Reason:** Fails for user with "modify", "get" and "delete", but not "add".
**Conditions:** RPC
**Status:** Implemented

**Number:** 12
**Priority:** High
**Reason:** Fails for user with "add" but not "delete".
**Conditions:** RPC
**Status:** Implemented

**Number:** 13
**Priority:** High
**Reason:** Fails for user with "delete" but not "add".
**Conditions:** RPC
**Status:** Implemented

**Number:** 14
**Priority:** High
**Reason:** Succeeds for user with "add" and "delete", when that user has non-name-based salt.
**Status:** Implemented

**Number:** 15

**Priority:** High

**Reason:** Fails if target principal name exists.

**Status:** Implemented

**Number:** 16

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 17

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

**Number:** 18

**Priority:** bug fix

**Reason:** Returns NO_RENAME_SALT when asked to rename a principal whose salt depends on the principal name.

**Status:** Implemented

# 8   ovsec_kadm_chpass_principal

## 8.1   Quality/history enforcement tests

This section lists a series of tests which will be run a number of times, with various parameter settings (e.g., which access bits user has, whether user connected with ADMIN_SERVICE or CHANGEPW_SERVICE, etc.). The table following the list of tests gives the various parameter settings under which the tests should be run, as well which should succeed and which should fail for each choice of parameter settings.

### 8.1.1   List of tests

The test number of each of these tests is an offset from the base given in the table below.

**Number:** 1

**Priority:** High

**Reason:** With history setting of 1, change password to itself.

**Number:** 2

**Reason:** With history setting of 2 but no password changes since principal creation, change password to itself.

**Number:** 3

**Reason:** With history setting of 2 and one password change since principal creation, change password to itself and directly previous password.

**Number:** 4

**Priority:** High

**Reason:** With a history setting of 3 and no password changes, change password to itself.

**Number:** 5

**Priority:** High

**Reason:** With a history setting of 3 and 1 password change, change password to itself or previous password.

**Number:** 6

**Priority:** High

**Reason:** With a history setting of 3 and 2 password changes, change password to itself and the two previous passwords.

**Number:** 7

**Priority:** High

**Reason:** Change to previously unused password when now - last_pwd_change < pw_min_life.

**Number:** 8

**Priority:** High

**Reason:** Change to previously unused password that doesn't contain enough character classes.

**Number:** 9

**Priority:** High

**Reason:** Change to previously unused password that's too short.

**Number:** 10

**Priority:** High

**Reason:** Change to previously unused password that's in the dictionary.

### 8.1.2 List of parameter settings

In the table below, "7 passes" means that test 7 above passes and the rest of the tests fail.

| Base | Modify access? | Own password? | Service | Pass/Fail |
|------|----------------|---------------|----------|-----------|
| 0 | No | Yes | ADMIN | all fail |
| 20 | No | Yes | CHANGEPW | all fail |
| 40 | No | No | ADMIN | all fail |
| 60 | No | No | CHANGEPW | all fail |
| 80 | Yes | Yes | ADMIN | 7 passes |
| 100 | Yes | Yes | CHANGEPW | all fail |
| 120 | Yes | No | ADMIN | 7 passes |
| 140 | Yes | No | CHANGEPW | all fail |

## 8.2 Other quality/history tests

**Number:** 161

**Priority:** High

**Reason:** With history of 1, can change password to anything other than itself that doesn't conflict with other quality rules.

**Number:** 162

**Reason:** With history of 2 and 2 password changes, can change password to original password.

**Number:** 163

**Priority:** High

**Reason:** With history of 3 and 3 password changes, can change password to original password.

**Number:** 164

**Priority:** High

**Reason:** Can change password when now - last_pwd_change > pw_min_life.

**Number:** 165

**Priority:** High

**Reason:** Can change password when it contains exactly the number of classes required by the policy.

**Number:** 166

**Priority:** High

**Reason:** Can change password when it is exactly the length required by the policy.

**Number:** 167

**Priority:** High

**Reason:** Can change password to a word that isn't in the dictionary.

## 8.3   Other tests

**Number:** 169

**Reason:** Fails for non-existent principal.

**Number:** 170

**Reason:** Fails for null password.

**Number:** 171

**Priority:** High

**Reason:** Fails for empty-string password.

**Number:** 172

**Priority:** High

**Reason:** Pw_expiration is set to now + max_pw_life if policy exists and has non-zero max_pw_life.

**Number:** 173

**Priority:** High

**Reason:** Pw_expiration is set to 0 if policy exists and has zero max_pw_life.

**Number:** 174

**Priority:** High

**Reason:** Pw_expiration is set to 0 if no policy.

**Number:** 175

**Priority:** High

**Reason:** KRB5_KDC_REQUIRES_PWCHANGE bit is cleared when password is successfully changed.

**Number:** 176
**Priority:** High
**Reason:** Fails for user with no access bits, on other's password.

**Number:** 177
**Priority:** High
**Reason:** Fails for user with "get" but not "modify" access, on other's password.

**Number:** 178
**Reason:** Fails for user with "delete" but not "modify" access, on other's password.

**Number:** 179
**Reason:** Fails for user with "add" but not "modify" access, on other's password.

**Number:** 180
**Reason:** Succeeds for user with "get" and "modify" access, on other's password.
**Status:** Implemented

**Number:** 180.5
**Priority:** High
**Reason:** Succeeds for user with "modify" but not "get" access, on other's password.
**Conditions:** RPC
**Status:** Implemented

**Number:** 180.625
**Priority:** High
**Reason:** Fails for user with modify when connecting with CHANGEPW_SERVICE on others password
**Conditions:** RPC
**Status:** Implemented

**Number:** 180.75
**Priority:** High
**Reason:** Fails for user with modify when connecting with CHANGEPW_SERVICE on other's password which has expired
**Conditions:** RPC
**Status:** Implemented

**Number:** 182
**Priority:** High
**Reason:** Can not change key of ovsec_adm/history principal.
**Status:** Implemented

**Number:** 183
**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 184

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

**Number:** 200

**Version:** KADM5_API_VERSION_2

**Reason:** Creates a key for the principal for each unique encryption type/salt type in use.

**Status:** Implemented

# 9  ovsec_kadm_chpass_principal_util

Rerun all the tests listed for ovsec_kadm_chpass_principal above in Section 8. Verify that they succeed and fail in the same circumstances. Also verify that in each failure case, the error message returned in msg_ret is as specified in the functional specification.

Also, run the following additional tests.

**Number:** 1

**Reason:** Null msg_ret is rejected.

**Number:** 2

**Priority:** High

**Reason:** New password is put into pw_ret, when it's prompted for.

**Number:** 3

**Priority:** High ReasonNew password is put into pw_ret, when it's supplied by the caller.

**Number:** 4

**Priority:** High

**Reason:** Successful invocation when pw_ret is null.

# 10  ovsec_kadm_randkey_principal

## 10.1  TOOSOON enforcement tests

This test should be run a number of times, as indicated in the table following it. The table also indicates the expected result of each run of the test.

**Reason:** Change key when now - last_pwd_change < pw_min_life.

### 10.1.1 List of parameter settings

| Number | Modify Access? | Own Key? | Service | Pass/Fail | Implemented? |
|--------|----------------|----------|---------|-----------|--------------|
| 1 | No | Yes | ADMIN | fail | Yes |
| 3 | No | Yes | CHANGEPW | fail | Yes |
| 5 | No | No | ADMIN | fail | |
| 7 | No | No | CHANGEPW | fail | |
| 9 | Yes | Yes | ADMIN | pass | |
| 11 | Yes | Yes | CHANGEPW | fail | |
| 13 | Yes | No | ADMIN | pass | Yes |
| 15 | Yes | No | CHANGEPW | fail | Yes |

## 10.2 Other tests

**Number:** 17

**Reason:** Fails if database not initialized.

**Number:** 18

**Reason:** Fails for non-existent principal.

**Number:** 19

**Reason:** Fails for null keyblock pointer.

**Number:** 20

**Priority:** High

**Reason:** Pw_expiration is set to now + max_pw_life if policy exists and has non-zero max_pw_life.

**Number:** 21

**Priority:** High

**Reason:** Pw_expiration is set to 0 if policy exists and has zero max_pw_life.

**Number:** 22

**Priority:** High

**Reason:** Pw_expiration is set to 0 if no policy.

**Number:** 23

**Priority:** High

**Reason:** KRB5_KDC_REQUIRES_PWCHANGE bit is cleared when key is successfully changed.

**Number:** 24

**Priority:** High

**Reason:** Fails for user with no access bits, on other's password.

**Number:** 25

**Priority:** High

**Reason:** Fails for user with "get" but not "modify" access, on other's password.

**V2 note:** Change-password instead of modify access.

**Number:** 26

**Reason:** Fails for user with "delete" but not "modify" access, on other's password.

**V2 note:** Change-password instead of modify access.

**Number:** 27

**Reason:** Fails for user with "add" but not "modify" access, on other's password.

**V2 note:** Change-password instead of modify access.

**Number:** 28

**Reason:** Succeeds for user with "get" and "modify" access, on other's password.

**Status:** Implemented

**V2 note:** Change-password instead of modify access.

**Number:** 28.25

**Priority:** High

**Reason:** Fails for user with get and modify access on others password When conneceted with CHANGEPW_SERVICE

**Status:** Implemented

**V2 note:** Change-password instead of modify access.

**Number:** 28.5

**Priority:** High

**Reason:** Succeeds for user with "modify" but not "get" access, on other's password.

**Status:** Implemented

**V2 note:** Change-password instead of modify access.

**Number:** 29

**Reason:** The new key that's assigned is truly random. XXX not sure how to test this.

**Number:** 30

**Reason:** Succeeds for own key, no other access bits when connecting with CHANGEPW service

**Status:** Implemented

**Number:** 31

**Reason:** Succeeds for own key, no other access bits when connecting with ADMIM service

**Status:** Implemented

**Number:** 32

**Reason:** Cannot change ovsec_adm/history key

**Status:** Implemented

**Number:** 33

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 34

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

**Number:** 100

**Version:** KADM5_API_VERSION_2

**Reason:** Returns a key for each unique encryption type specified in the keysalts.

# 11  ovsec_kadm_get_principal

**Number:** 1

**Reason:** Fails for null ent.

**Status:** Implemented

**Number:** 2

**Reason:** Fails for non-existent principal.

**Status:** Implemented

**Number:** 3

**Priority:** High

**Reason:** Fails for user with no access bits, retrieving other principal.

**Conditions:** RPC

**Status:** Implemented

**Number:** 4

**Priority:** High

**Reason:** Fails for user with "add" but not "get", getting principal other than his own, using ADMIN_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 5

**Reason:** Fails for user with "modify" but not "get", getting principal other than his own, using ADMIN_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 6

**Reason:** Fails for user with "delete" but not "get", getting principal other than his own, using ADMIN_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 7

**Reason:** Fails for user with "delete" but not "get", getting principal other than his own, using CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 8
**Priority:** High
**Reason:** Fails for user with "get", getting principal other than his own, using CHANGEPW_SERVICE.
**Conditions:** RPC
**Status:** Implemented

**Number:** 9
**Priority:** High
**Reason:** Succeeds for user without "get", retrieving self, using ADMIN_SERVICE.
**Conditions:** RPC
**Status:** Implemented

**Number:** 10
**Reason:** Succeeds for user without "get", retrieving self, using CHANGEPW_SERVICE.
**Status:** Implemented

**Number:** 11
**Reason:** Succeeds for user with "get", retrieving self, using ADMIN_SERVICE.
**Status:** Implemented

**Number:** 12
**Reason:** Succeeds for user with "get", retrieving self, using CHANGEPW_SERVICE.
**Status:** Implemented

**Number:** 13
**Priority:** High
**Reason:** Succeeds for user with "get", retrieving other user, using ADMIN_SERVICE.
**Status:** Implemented

**Number:** 14
**Reason:** Succeeds for user with "get" and "modify", retrieving other principal, using ADMIN_SERVICE.
**Status:** Implemented

**Number:** 15
**Priority:** High
**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in
**Status:** Implemented

**Number:** 16
**Priority:** Low
**Reason:** Connects to correct server when mutliple handles exist
**Conditions:** RPC

**Number:** 100
**Version:** KADM5_API_VERSION_2

**Reason:** If KADM5_PRINCIPAL_NORMAL_MASK is specified, the key_data and tl_data fields are NULL/zero.

**Status:** Implemented

**Number:** 101

**Version:** KADM5_API_VERSION_2

**Reason:** If KADM5_KEY_DATA is specified, the key_data fields contain data but the contents are all NULL.

**Conditions:** RPC

**Status:** Implemented

**Number:** 102

**Version:** KADM5_API_VERSION_2

**Reason:** If KADM5_KEY_DATA is specified, the key_data fields contain data and the contents are all non-NULL.

**Conditions:** local

**Status:** Implemented

**Number:** 103

**Version:** KADM5_API_VERSION_2

**Reason:** If KADM5_TL_DATA is specified, the tl_data field contains the correct tl_data and no entries whose type is less than 256.

**Status:** Implemented

# 12   ovsec_kadm_create_policy

**Number:** 1

**Reason:** Fails for mask with undefined bit set.

**Status:** Implemented - untested

**Number:** 2

**Priority:** High

**Reason:** Fails if caller connected with CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 3

**Reason:** Fails for mask without POLICY bit set.

**Status:** Implemented - untested

**Number:** 4

**Reason:** Fails for mask with REF_COUNT bit set.

**Status:** Implemented

**Number:** 5

**Reason:** Fails for invalid policy name.

**Status:** Implemented - untested

**Number:** 6

**Priority:** High

**Reason:** Fails for existing policy name.

**Status:** Implemented

**Number:** 7

**Reason:** Fails for null policy name.

**Status:** Implemented - untested

**Number:** 8

**Priority:** High

**Reason:** Fails for empty-string policy name.

**Status:** Implemented

**Number:** 9

**Priority:** High

**Reason:** Accepts 0 for pw_min_life.

**Status:** Implemented

**Number:** 10

**Priority:** High

**Reason:** Accepts non-zero for pw_min_life.

**Status:** Implemented

**Number:** 11

**Priority:** High

**Reason:** Accepts 0 for pw_max_life.

**Status:** Implemented

**Number:** 12

**Priority:** High

**Reason:** Accepts non-zero for pw_max_life.

**Status:** Implemented

**Number:** 13

**Priority:** High

**Reason:** Rejects 0 for pw_min_length.

**Status:** Implemented

**Number:** 14

**Priority:** High

**Reason:** Accepts non-zero for pw_min_length.

**Status:** Implemented

**Number:** 15

**Priority:** High
**Reason:** Rejects 0 for pw_min_classes.
**Status:** Implemented

**Number:** 16
**Priority:** High
**Reason:** Accepts 1 for pw_min_classes.
**Status:** Implemented

**Number:** 17
**Priority:** High
**Reason:** Accepts 4 for pw_min_classes.
**Status:** Implemented

**Number:** 18
**Priority:** High
**Reason:** Rejects 5 for pw_min_classes.
**Status:** Implemented

**Number:** 19
**Priority:** High
**Reason:** Rejects 0 for pw_history_num.
**Status:** Implemented

**Number:** 20
**Priority:** High
**Reason:** Accepts 1 for pw_history_num.
**Status:** Implemented

**Number:** 21
**Priority:** High
**Reason:** Accepts 10 for pw_history_num.
**Status:** Implemented

**Number:** 21.5
**Reason:** Rejects 11 for pw_history_num.
**Status:** Implemented - untested

**Number:** 22
**Priority:** High
**Reason:** Fails for user with no access bits.
**Conditions:** RPC
**Status:** Implemented

**Number:** 23

**Priority:** High

**Reason:** Fails for user with "get" but not "add".

**Conditions:** RPC

**Status:** Implemented

**Number:** 24

**Reason:** Fails for user with "modify" but not "add."

**Conditions:** RPC

**Status:** Implemented - untested

**Number:** 25

**Reason:** Fails for user with "delete" but not "add."

**Conditions:** RPC

**Status:** Implemented - untested

**Number:** 26

**Priority:** High

**Reason:** Succeeds for user with "add."

**Status:** Implemented

**Number:** 27

**Reason:** Succeeds for user with "get" and "add."

**Status:** Implemented - untested

**Number:** 28

**Reason:** Rejects null policy argument.

**Status:** Implemented - untested

**Number:** 29

**Reason:** Rejects pw_min_life greater than pw_max_life.

**Number:** 30

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 31

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

# 13   ovsec_kadm_delete_policy

**Number:** 1
**Reason:** Fails for null policy name.

**Number:** 2
**Priority:** High
**Reason:** Fails for empty-string policy name.
**Status:** Implemented

**Number:** 3
**Reason:** Fails for non-existent policy name.

**Number:** 4
**Reason:** Fails for bad policy name.

**Number:** 5
**Priority:** High
**Reason:** Fails if caller connected with CHANGEPW_SERVICE.
**Conditions:** RPC
**Status:** Implemented

**Number:** 6
**Priority:** High
**Reason:** Fails for user with no access bits.
**Conditions:** RPC
**Status:** Implemented

**Number:** 7
**Priority:** High
**Reason:** Fails for user with "add" but not "delete".
**Conditions:** RPC
**Status:** Implemented

**Number:** 8
**Reason:** Fails for user with "modify" but not "delete".
**Conditions:** RPC

**Number:** 9
**Reason:** Fails for user with "get" but not "delete."
**Conditions:** RPC

**Number:** 10
**Priority:** High
**Reason:** Succeeds for user with only "delete".

**Status:** Implemented

**Number:** 11
**Reason:** Succeeds for user with "delete" and "add".

**Number:** 12
**Priority:** High
**Reason:** Fails for policy with non-zero reference count.
**Status:** Implemented

**Number:** 13
**Priority:** High
**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in
**Status:** Implemented

**Number:** 14
**Priority:** Low
**Reason:** Connects to correct server when mutliple handles exist
**Conditions:** RPC

# 14  ovsec_kadm_modify_policy

**Number:** 1
**Reason:** Fails for mask with undefined bit set.
**Conditions:** RPC

**Number:** 2
**Priority:** High
**Reason:** Fails if caller connected with CHANGEPW_SERVICE.
**Status:** Implemented

**Number:** 3
**Reason:** Fails for mask with POLICY bit set.

**Number:** 4
**Reason:** Fails for mask with REF_COUNT bit set.
**Status:** Implemented

**Number:** 5
**Reason:** Fails for invalid policy name.

**Number:** 6
**Reason:** Fails for non-existent policy name.

**Number:** 7
**Reason:** Fails for null policy name.

**Number:** 8
**Priority:** High
**Reason:** Fails for empty-string policy name.
**Status:** Implemented

**Number:** 9
**Priority:** High
**Reason:** Accepts 0 for pw_min_life.
**Status:** Implemented

**Number:** 10
**Priority:** High
**Reason:** Accepts non-zero for pw_min_life.
**Status:** Implemented

**Number:** 11
**Priority:** High
**Reason:** Accepts 0 for pw_max_life.
**Status:** Implemented

**Number:** 12
**Priority:** High
**Reason:** Accepts non-zero for pw_max_life.
**Status:** Implemented

**Number:** 13
**Priority:** High
**Reason:** Accepts 0 for pw_min_length.
**Status:** Implemented

**Number:** 14
**Priority:** High
**Reason:** Accepts non-zero for pw_min_length.
**Status:** Implemented

**Number:** 15
**Priority:** High
**Reason:** Rejects 0 for pw_min_classes.
**Status:** Implemented

**Number:** 16
**Priority:** High
**Reason:** Accepts 1 for pw_min_classes.
**Status:** Implemented

**Number:** 17

**Priority:** High

**Reason:** Accepts 4 for pw_min_classes.

**Status:** Implemented

**Number:** 18

**Priority:** High

**Reason:** Rejects 5 for pw_min_classes.

**Status:** Implemented

**Number:** 19

**Priority:** High

**Reason:** Rejects 0 for pw_history_num.

**Status:** Implemented

**Number:** 20

**Priority:** High

**Reason:** Accepts 1 for pw_history_num.

**Status:** Implemented

**Number:** 21

**Priority:** High

**Reason:** Accepts 10 for pw_history_num.

**Status:** Implemented

**Number:** 22

**Priority:** High

**Reason:** Fails for user with no access bits.

**Conditions:** RPC

**Status:** Implemented

**Number:** 23

**Priority:** High

**Reason:** Fails for user with "get" but not "modify".

**Conditions:** RPC

**Status:** Implemented

**Number:** 24

**Reason:** Fails for user with "add" but not "modify."

**Conditions:** RPC

**Number:** 25

**Reason:** Fails for user with "delete" but not "modify."

**Conditions:** RPC

**Number:** 26
**Priority:** High
**Reason:** Succeeds for user with "modify."
**Status:** Implemented

**Number:** 27
**Reason:** Succeeds for user with "get" and "modify."

**Number:** 28
**Reason:** Rejects null policy argument.

**Number:** 29
**Reason:** Rejects change which makes pw_min_life greater than pw_max_life.

**Number:** 30
**Priority:** High
**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in
**Status:** Implemented

**Number:** 31
**Priority:** Low
**Reason:** Connects to correct server when mutliple handles exist
**Conditions:** RPC

# 15 ovsec_kadm_get_policy

**Number:** 1
**Reason:** Fails for null policy.

**Number:** 2
**Reason:** Fails for invalid policy name.

**Number:** 3
**Priority:** High
**Reason:** Fails for empty-string policy name.
**Status:** Implemented

**Number:** 4
**Reason:** Fails for non-existent policy name.

**Number:** 5
**Reason:** Fails for null ent.

**Number:** 6
**Priority:** High
**Reason:** Fails for user with no access bits trying to get other's policy, using ADMIN_SERVICE.

**Conditions:** RPC
**Status:** Implemented

**Number:** 7
**Priority:** High
**Reason:** Fails for user with "add" but not "get" trying to get other's policy, using ADMIN_SERVICE.
**Conditions:** RPC
**Status:** Implemented

**Number:** 8
**Reason:** Fails for user with "modify" but not "get" trying to get other's policy, using ADMIN_SERVICE.
**Conditions:** RPC

**Number:** 9
**Reason:** Fails for user with "delete" but not "get" trying to get other's policy, using ADMIN_SERVICE.
**Conditions:** RPC

**Number:** 10
**Reason:** Fails for user with "delete" but not "get" trying to get other's policy, using CHANGEPW_SERVICE.
**Conditions:** RPC

**Number:** 11
**Priority:** High
**Reason:** Succeeds for user with only "get", trying to get own policy, using ADMIN_SERVICE.
**Status:** Implemented

**Number:** 12
**Priority:** High
**Reason:** Succeeds for user with only "get", trying to get own policy, using CHANGEPW_SERVICE.
**Status:** Implemented

**Number:** 13
**Reason:** Succeeds for user with "add" and "get", trying to get own policy, using ADMIN_SERVICE.

**Number:** 14
**Reason:** Succeeds for user with "add" and "get", trying to get own policy, using CHANGEPW_SERVICE.

**Number:** 15
**Reason:** Succeeds for user without "get", trying to get own policy, using ADMIN_SERVICE.

**Number:** 16
**Priority:** High
**Reason:** Succeeds for user without "get", trying to get own policy, using CHANGEPW_SERVICE.
**Status:** Implemented

**Number:** 17

**Priority:** High

**Reason:** Succeeds for user with "get", trying to get other's policy, using ADMIN_SERVICE.

**Status:** Implemented

**Number:** 18

**Priority:** High

**Reason:** Fails for user with "get", trying to get other's policy, using CHANGEPW_SERVICE.

**Conditions:** RPC

**Status:** Implemented

**Number:** 19

**Reason:** Succeeds for user with "modify" and "get", trying to get other's policy, using ADMIN_SERVICE.

**Number:** 20

**Reason:** Fails for user with "modify" and "get", trying to get other's policy, using CHANGEPW_SERVICE.

**Number:** 21

**Priority:** High

**Reason:** Returns BAD_SERVER_HANDLE when a null server handle is passed in

**Status:** Implemented

**Number:** 22

**Priority:** Low

**Reason:** Connects to correct server when mutliple handles exist

**Conditions:** RPC

# 16   ovsec_kadm_free_principal_ent

In addition to the tests listed here, a memory-leak detector such as TestCenter, Purify or dbmalloc should be used to verify that the memory freed by this function is really freed.

**Number:** 1

**Reason:** Null princ succeeds.

**Number:** 2

**Reason:** Non-null princ succeeds.

# 17   ovsec_kadm_free_policy_ent

In addition to the tests listed here, a memory-leak detector such as TestCenter, Purify or dbmalloc should be used to verify that the memory freed by this function is really freed.

**Number:** 1

**Reason:** Null policy succeeds.

**Number:** 2

**Reason:** Non-null policy succeeds.

# 18 ovsec_kadm_get_privs

**Number:** 1

**Reason:** Fails for null pointer argument.

This test should be run with the 16 possible combinations of access bits (since there are 4 access bits, there are $2^4 = 16$ possible combinations of them):

**Number:** 2

**Priority:** High

**Reason:** Returns correct bit mask for access bits of user.

**Conditions:** RPC

This test should be run locally:

**Number:** 3

**Priority:** High

**Reason:** Returns 0x0f.

**Conditions:** local