

# 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 should 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 the structure than 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 than supported

**Conditions:** RPC

**Status:** Implemented

**Number:** 40

**Priority:** High

**Reason:** Return NEW\_SERVER\_API\_VERSION when using a newer api version number than 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 succeeds, 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 multiple 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 mutiple 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 mutliples 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 multiple 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  $\text{now} - \text{last\_pwd\_change} < \text{pw\_min\_life}$ .

**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	CHANGE PW	all fail
40	No	No	ADMIN	all fail
60	No	No	CHANGE PW	all fail
80	Yes	Yes	ADMIN	7 passes
100	Yes	Yes	CHANGE PW	all fail
120	Yes	No	ADMIN	7 passes
140	Yes	No	CHANGE PW	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  $\text{now} - \text{last\_pwd\_change} > \text{pw\_min\_life}$ .

**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  $\text{now} + \text{max\_pw\_life}$  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  $\text{now} - \text{last\_pwd\_change} < \text{pw\_min\_life}$ .

### 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 connected 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 multiple 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 CHANGE PW\_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 mutiple 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 mutiple 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 multiple 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