- Open Vulnerability and Assessment Language - Element Dictionary

- Schema: MacOS Definition
- Version: 5.2
- Release Date: 31 January 2007

The following is a description of the elements, types, and attributes that compose the MacOS specific tests found in Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standard test element defined in the Core Definition Schema. Through extension, each test inherits a set of elements and attributes that are shared amongst all OVAL tests. Each test is described in detail and should provide the information necessary to understand what each element and attribute represents. This document is intended for developers and assumes some familiarity with XML. A high level description of the interaction between the different tests and their relationship to the Core Definition Schema is not outlined here.

The MacOS Definition Schema was initially developed by The Center for Internet Security. Many thanks to their contributions to OVAL and the security community.

The OVAL Schema is maintained by The Mitre Corporation and developed by the public OVAL Community. For more information, including how to get involved in the project and how to submit change requests, please visit the OVAL website at http://oval.mitre.org.

< accountinfo_test >

User account information (username, uid, gid, etc.) See netinfo(5) for field information, niutil(1) for retrieving it. We may need/want to add in data elements for things like authentication_authority, generateduid, mcx_settings (restricted account settings).

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

< accountinfo_object >

The accountinfo_object element is used by an accountinfo_test to define the object(s) to be evaluated. This object extends the standard ObjectType as definied in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

An accountinfo_object consists of a single username that identifies the account from which to gather information.

Child Elements	Type	MinOccurs	MaxOccurs
username	oval-def:EntityObjectStringType	1	1

< accountinfo_state >

The accountinfo_state element defines the different information that can be used to evaluate the specified accounts. Please refer to the individual elements in the schema for more details about what each represents.

Child Elements	Type	MinOccurs	MaxOccurs
username	oval-def:EntityStateStringType	0	1
password	oval-def:EntityStateStringType	0	1
uid	oval-def:EntityStateIntType	0	1
gid	oval-def:EntityStateIntType	0	1
realname	oval-def:EntityStateStringType	0	1
home_dir	oval-def:EntityStateStringType	0	1
login_shell	oval-def:EntityStateStringType	0	1

< inetlisteningservers_test >

This test's purpose is generally used to check if a program is listening on the network, either for a new connections or as part of an ongoing connection. It is generally speaking the parsed output of running the command netstat -tuwlnpe with root privilege.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

< inetlisteningservers_object >

The inetlisteningservers_object element is used by an inetlisteningserver test to define the object to be evaluated. Each object extends the standard ObjectType as definied in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Child Elements	Type	MinOccurs	MaxOccurs
program_name	oval-def:EntityObjectStringType	1	1

< inetlisteningservers_state >

The inetlisteningservers_state element defines the different information that can be used to evaluate the specified inet listening server. This includes the local address, foreign address, port information, and process id. Please refer to the individual elements in the schema for more details about what each represents.

Child Elements	Type	MinOccurs	MaxOccurs
program_name	oval-def:EntityStateStringType	0	1
local_address	oval-def:EntityStateStringType	0	1

local_full_address	oval-def:EntityStateStringType	0	1
local_port	oval-def:EntityStateStringType	0	1
foreign_address	oval-def:EntityStateStringType	0	1
foreign_full_address	oval-def:EntityStateStringType	0	1
foreign_port	oval-def:EntityStateStringType	0	1
pid	oval-def:EntityStateIntType	0	1
protocol	oval-def:EntityStateStringType	0	1
user_id	oval-def:EntityStateStringType	0	1

< nvram_test >

This test pulls data from the 'nvram -p' output.

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

< nvram_object >

The nvram_object element is used by a nvram test to define the object to be evaluated. Each object extends the standard ObjectType as definied in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Child Elements	Type	MinOccurs	MaxOccurs
nvram_var	oval-def:EntityObjectStringType	1	1

< nvram_state >

This test pulls data from the 'nvram -p' output.

Child Elements	Type	MinOccurs	MaxOccurs
nvram_var	oval-def:EntityStateStringType	0	1
nvram_value	oval-def:EntityStateStringType	0	1

< pwpolicy_test >

This test pulls data from the 'pwpolicy -getpolicy' output. The actual values get stored under /var/db/netinfo/local.nidb/ in a Store.# file. Is this test actually needed, or can the text file content test be used instead?

Child Elements	Type	MinOccurs	MaxOccurs
object	oval-def:ObjectRefType	1	1
state	oval-def:StateRefType	0	1

< pwpolicy_object >

The pwpolicy_object element is used by a pwpolicy test to define the object to be evaluated. Each object extends the standard ObjectType as definied in the oval-definitions-schema and one should refer to the ObjectType description for more information. The common set element allows complex objects to be created using filters and set logic. Again, please refer to the description of the set element in the oval-definitions-schema.

Child Elements	Type	MinOccurs	MaxOccurs
username	oval-def:EntityObjectStringType	1	1
userpass	oval-def:EntityObjectStringType	1	1
directory_node	oval-def:EntityObjectStringType	1	1

< pwpolicy_state >

Child Elements	Type	MinOccurs	MaxOccurs
username	oval-def:EntityStateStringType	0	1
userpass	oval-def:EntityStateStringType	0	1
directory_node	oval-def:EntityStateStringType	0	1
maxChars	oval-def:EntityStateIntType	0	1
maxFailedLoginAttempts	oval-def:EntityStateIntType	0	1
minChars	oval-def:EntityStateIntType	0	1
passwordCannotBeName	oval-def:EntityStateBoolType	0	1
requiresAlpha	oval-def:EntityStateBoolType	0	1
requiresNumeric	oval-def:EntityStateBoolType	0	1