VMware SCAP Edit

User Guide

Revision 1.0 – 11-Jul-2016

Table of Contents

[Introduction 3](#_Toc452904518)

[Installation and getting started 3](#_Toc452904519)

[System Requirements 3](#_Toc452904520)

[Installation 4](#_Toc452904521)

[Using the tool 4](#_Toc452904522)

[Creating a new OVAL file 5](#_Toc452904523)

[Adding a new OVAL definition 10](#_Toc452904524)

[Opening an existing OVAL file 46](#_Toc452904525)

[Creating XCCDF file from an existing OVAL file 48](#_Toc452904526)

[References 51](#_Toc452904527)

# Introduction

VMware SCAP Edit is an updated version of the [Enhanced SCAP Content Editor](http://www.g2-inc.com/escape) tool by G2, Inc. The modified tool builds on similar capabilities but few subtle differences required to build [OVAL](https://github.com/OVALProject/Language) content based on latest OVAL schema.

The major changes done by VMware are as below:

* Added support for OVAL 5.11 for Independent, Unix, Linux and Windows schemas
* Added support for [XCCDF 1.2](https://scap.nist.gov/specifications/xccdf/#resource-1.2) creation from OVAL
* Refreshed the tool with modern UI
* Dropped broken capabilities from previous versions of the tool
* Dropped obsolete schemas and all other seldom used features
* Updated libraries to latest versions
* Updated [CPE version to 2.3](https://nvd.nist.gov/cpe.cfm)
* Restructured the code
* Removed obsolete and unneeded libraries

These changes make the tool simple and intuitive to build OVAL content specially aligned with the latest 5.11 OVAL schema. Also, it can now create XCCDF 1.2 content directly from OVAL file.

*Note: This tool and the documentation below assumes that you are fairly aware of* [*SCAP*](https://scap.nist.gov/revision/1.2/)*, OVAL, XCCDF and the surrounding scheme of things. If you are not, please take a moment and familiarize yourself with these before proceeding further.*

# Installation and getting started

## System Requirements

* A Java Runtime Environment ([JRE](http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html)) preferably 1.8 or later.
* 1 GB of memory.
* Supported OS – Microsoft Windows and Linux (with support for JRE).

## Installation

1. Extract the **VMware SCAP Edit 1.0.0.zip** file.
2. Navigate to the extracted directory.
3. If you are running a Microsoft Windows system, execute **startEditor.bat** file. This will launch the tool. If you are running a Linux system, execute **startEditor.sh** file to launch the tool.

These scripts will call Java with the appropriate arguments. These can be edited if you need to give Java more memory or set some other system property. Editing these files is only recommended for advanced users.

# Using the tool

Launch the tool. You would see the tool interface as below:



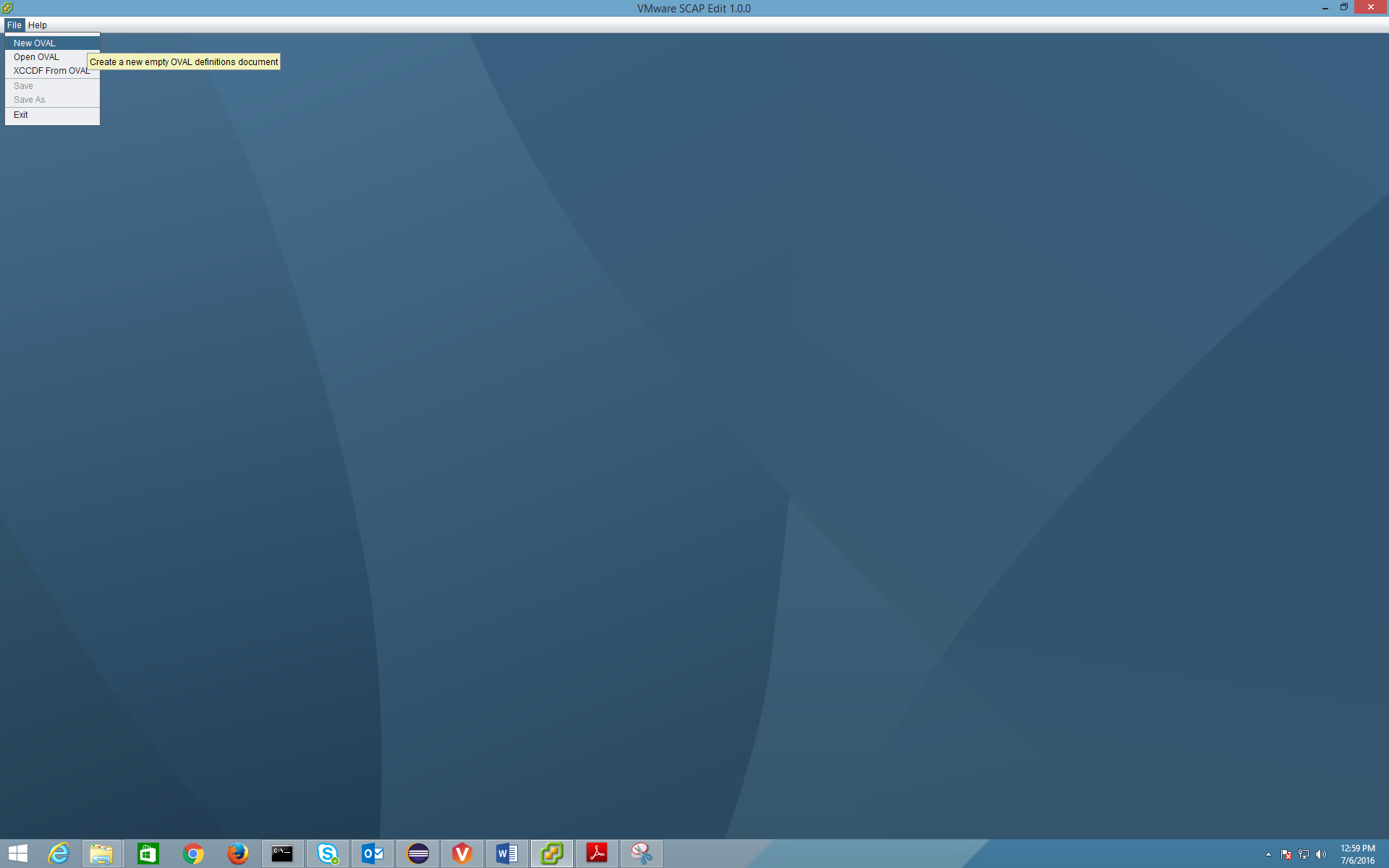
As you immediately notice, the tool has been simplified from its parent version. It now just has a file menu that offers you below core capabilities:

1. Creating a new OVAL file
2. Opening an existing OVAL file
3. Creating a XCCDF file from an existing OVAL file

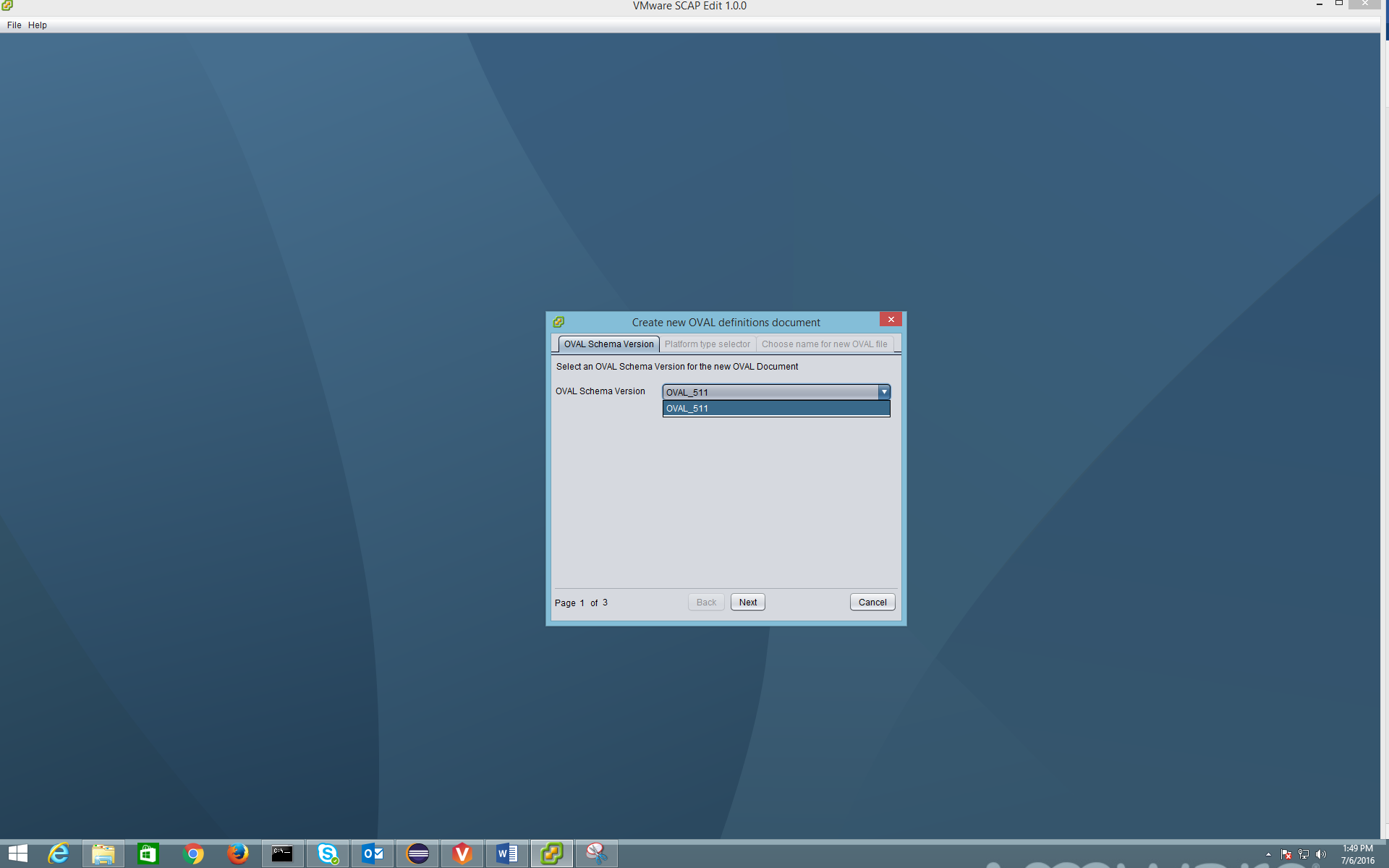
All other functionalities from the parent version have been taken out. Now, let us see each of the three functionalities in depth.

## Creating a new OVAL file

Launch the tool. Navigate to **File 🡪 New OVAL**.



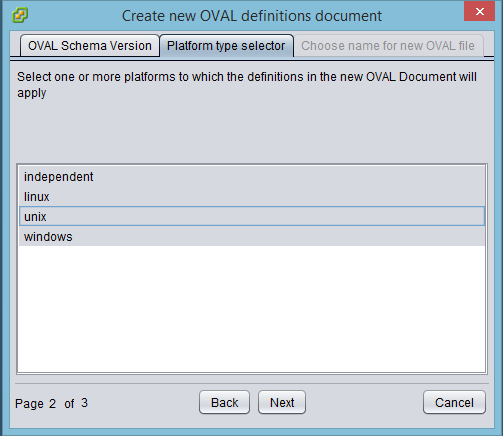
This opens a new dialog box with default OVAL 5.11 schema chosen. We have removed all other obsolete schema.



Click **Next**.

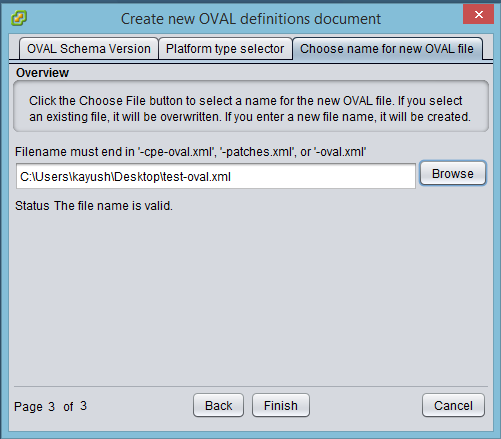
Now, select the target platform. You may now only choose between Independent, Linux, Unix and Windows. We have dropped the support for any other platform.

For example, let us choose **Unix**.



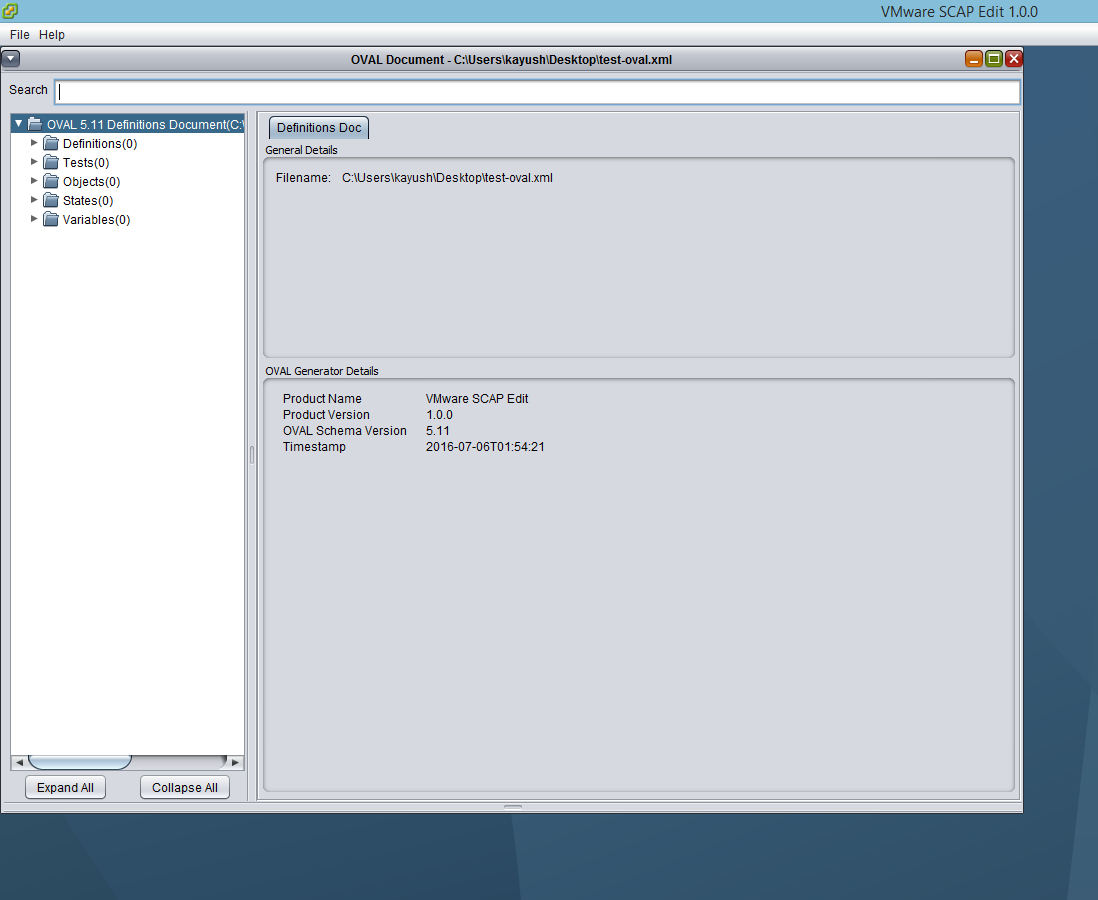
Click **Next**.

Finally, choose the file name.



Click **Finish**.

This creates a new and blank OVAL file.



This tool helps you define various OVAL components:

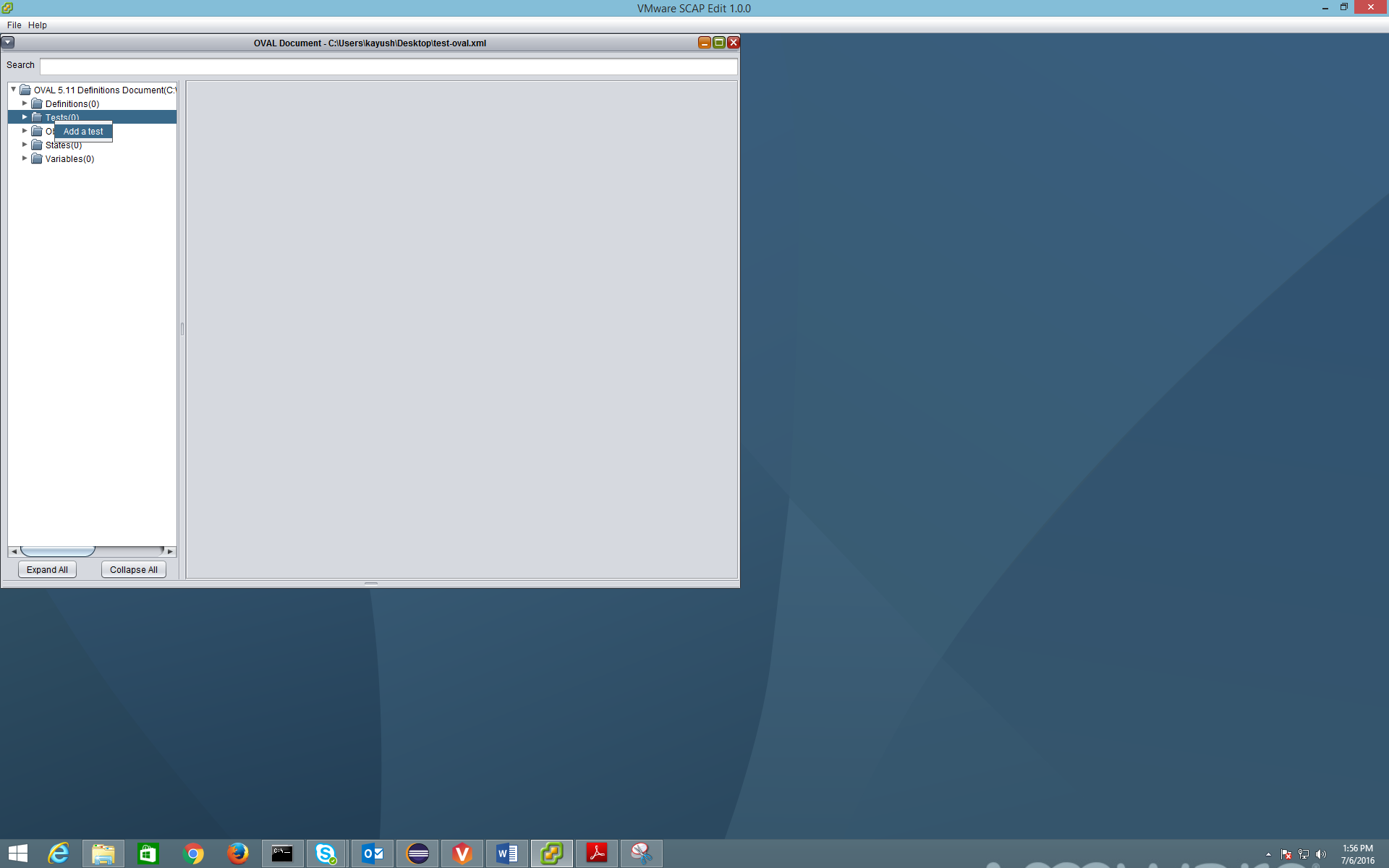
* Definitions
* Tests
* Objects
* States and
* Variables

You can define those components individually and then link them up together later. But, for this example, let us walk through a typical flow in creating an OVAL definition.

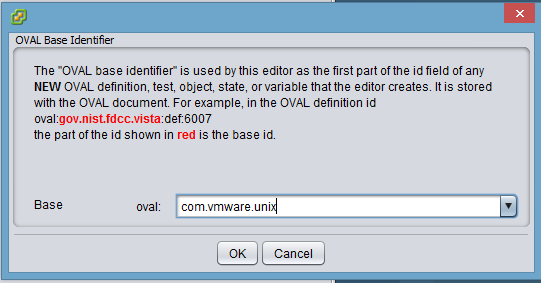
### Adding a new OVAL definition

Let us take an example that we want to write an OVAL definition that verifies that the /etc/passwd file is owned and group-owned by root and has permissions of 644.

Start with adding a Test. Right click on **Tests** and then click **Add Test**.



For your first test only, you will have to define OVAL base identifier.



For this example, let us set the base identifier to **com.vmware.unix**. You can set this up as you would require.

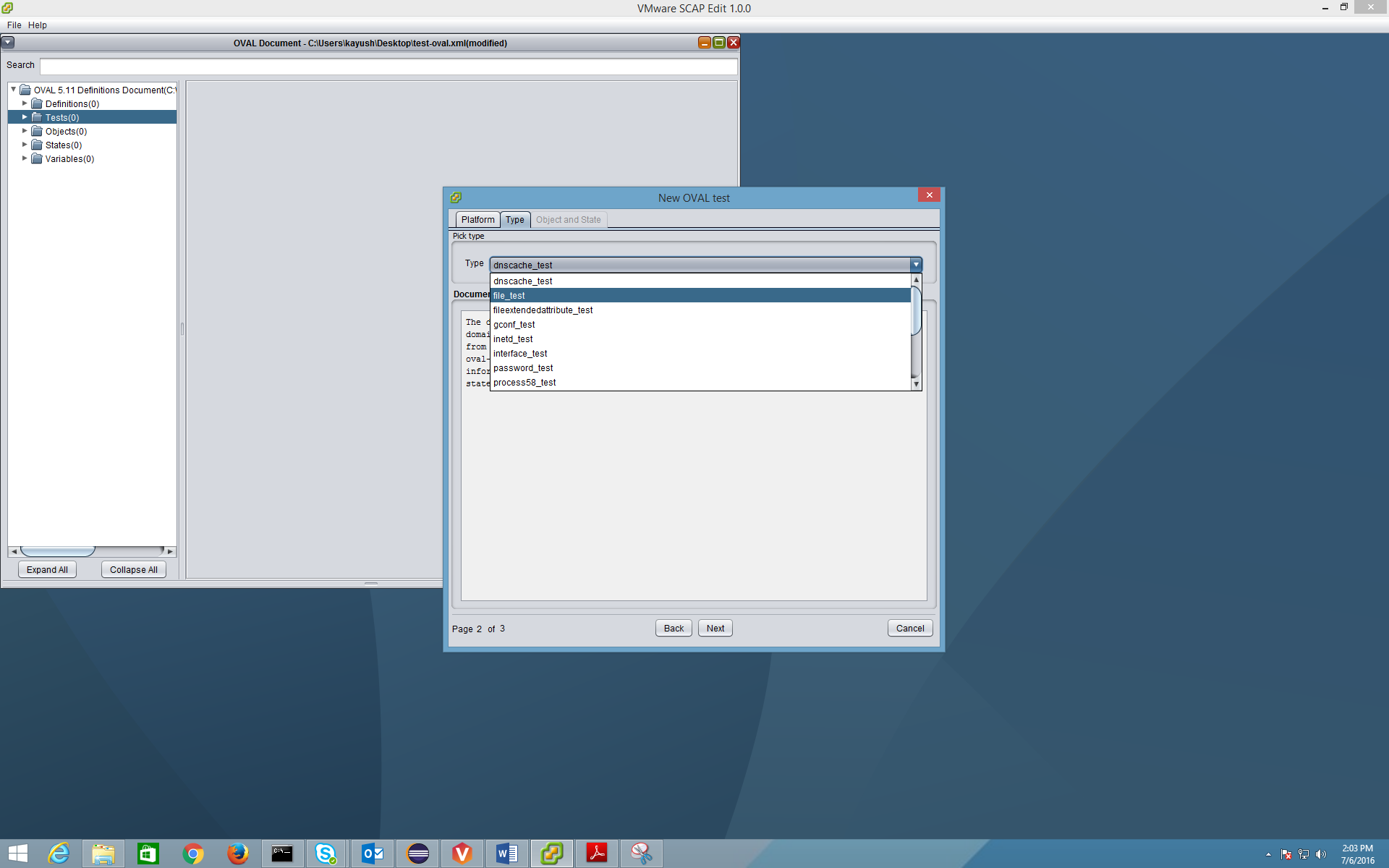
Click **OK**.

Now, choose the target platform for which you are writing this test. Let us choose **Unix** here.



Click **Next**.

Now choose Test type or OVAL probe against which you want to define the test. Check out the [references](#_References) to read about these in detail.



For this example, let us choose **file\_test**. Using this test type we can check various attributes on files such as ownership and permissions. Click **Next**.

As you know, each test is comprised of objects and states. On this screen, we will define objects and states to match our requirements (checking /etc/passwd file for ownership and group-ownership to be root and permissions of 644).

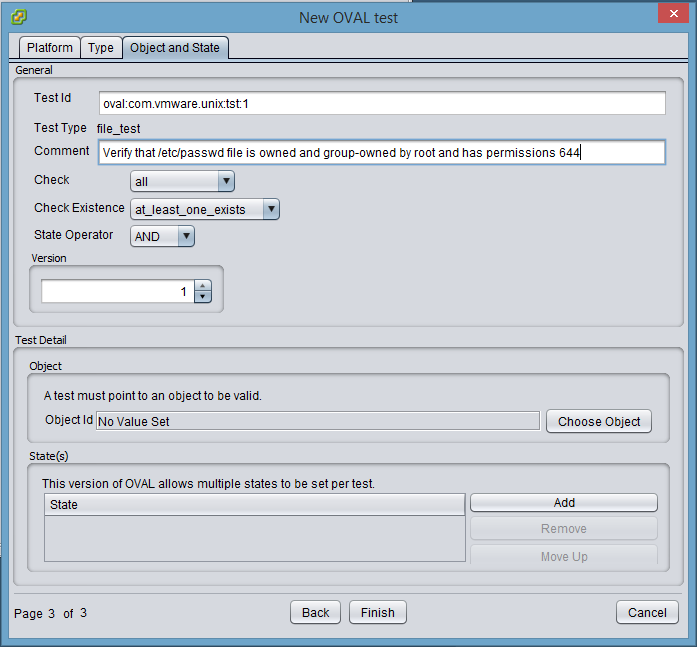


As you notice above, the base identifier is preset. Also, you don’t have to manage the ids yourself once you define the base identifier. The tool keeps incrementing the ids as you add more tests. The same holds true for any other OVAL components such as definitions, objects, states and variables that you write. All the ids are taken care of by the tool.

Also, all the OVAL defaults for various fields such as Check, Check Existence and State Operator are preset. You can use the drop down to pick and choose the various options as matching your requirements.

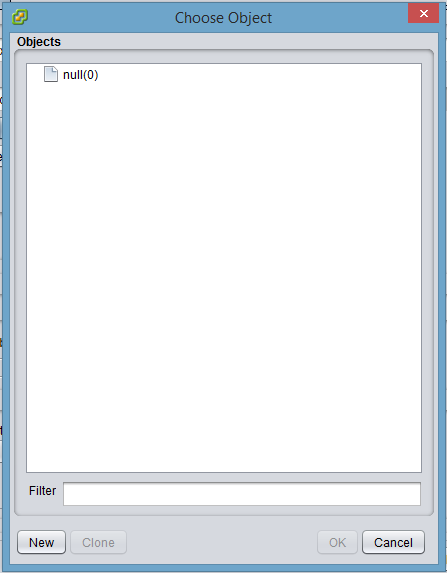
Now, let us go ahead and define the test as needed by our requirement.

We start by giving a **Comment**.



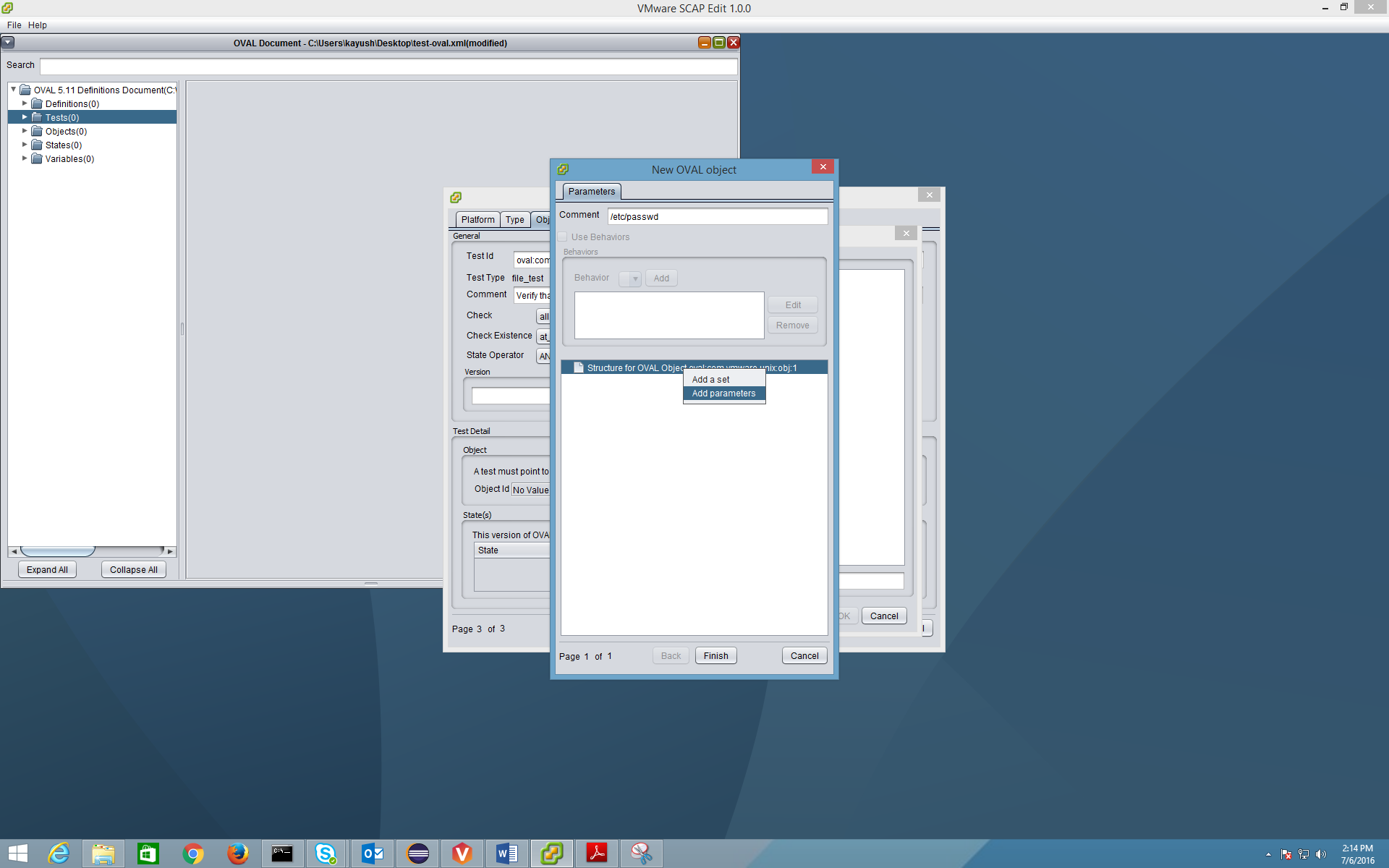
Now click on **Choose Object**.

This opens up Object wizard.

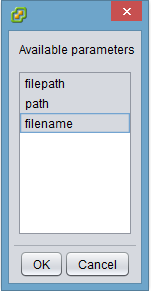


Click **New**.

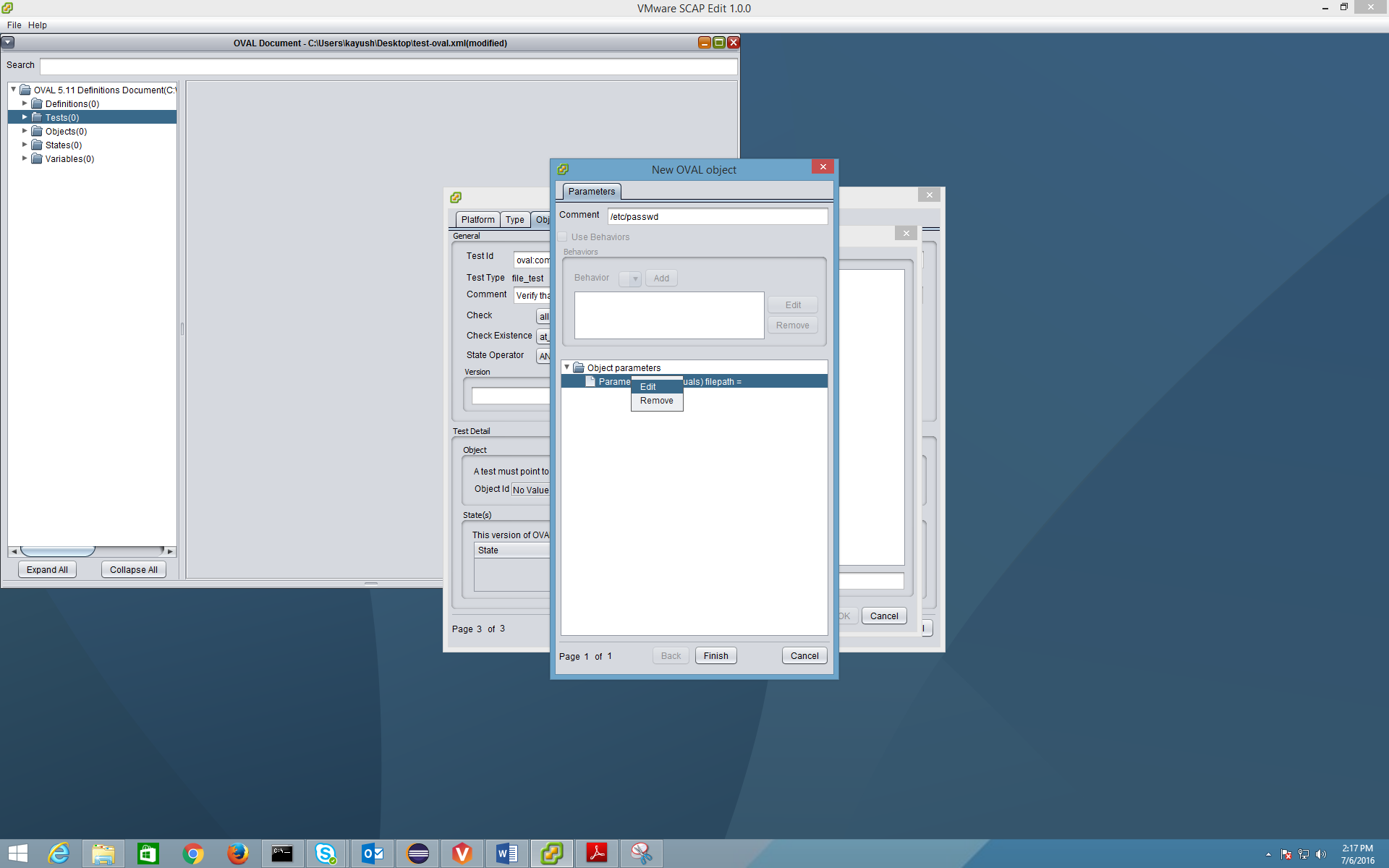
This opens up New Object wizard. Give a comment. Right click on the below shown area to add parameters.



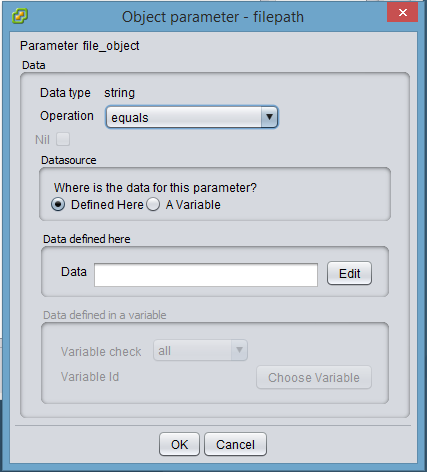
Since, we chose test type as file\_test, only below corresponding object parameters are available.



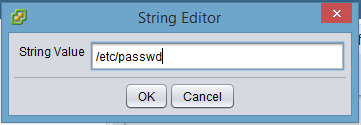
Here, let us choose to give absolute path of the file. In our example, it is /etc/passwd. Click **OK**.



Right click and select **Edit** to set the value for filepath object parameter.

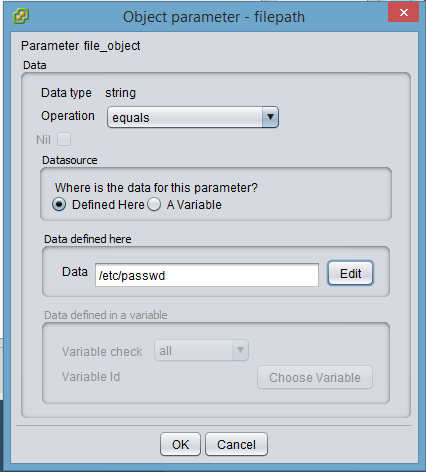


In this wizard, click on **Edit** and define the parameter value.



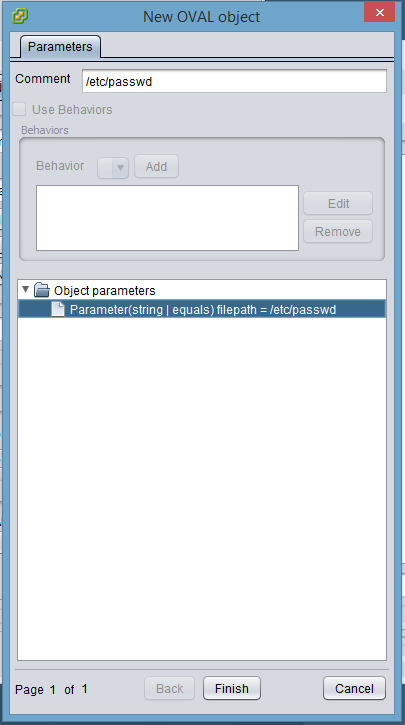
Click **OK**.

The value gets set for filepath.



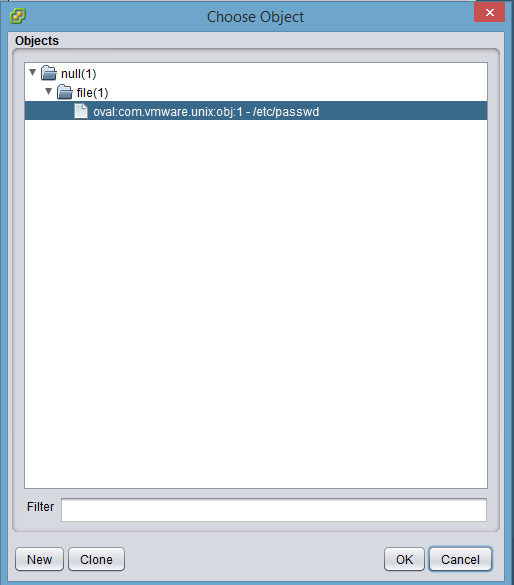
Click **OK**.

The object is now defined.



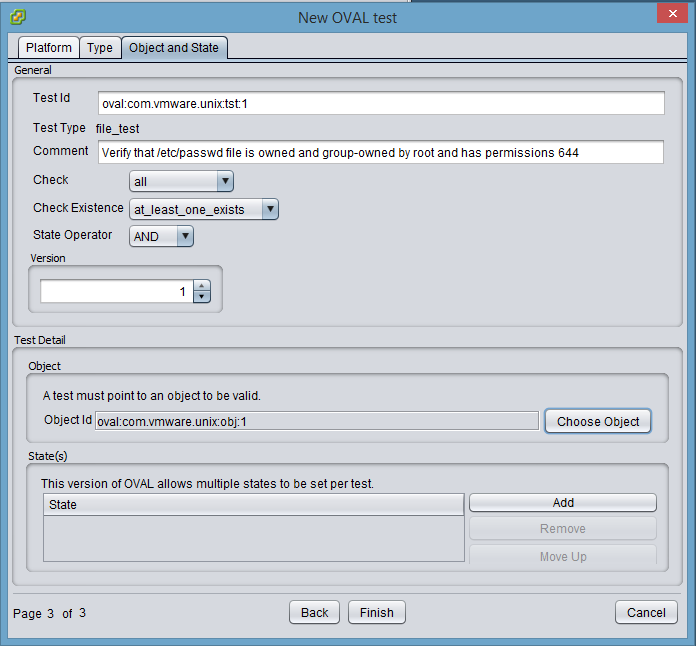
Click **Finish**.

Object is now chosen for the corresponding test.



Click **OK**.

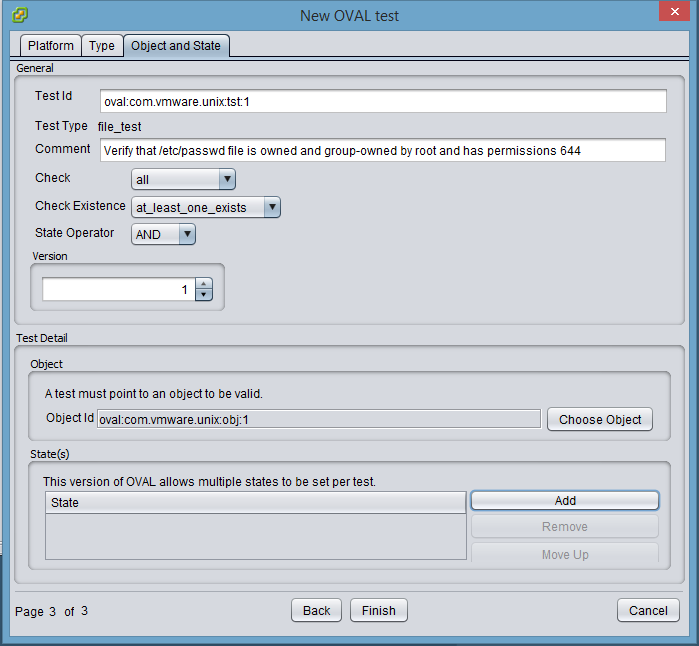
You will now see that the object id is populated in the test.



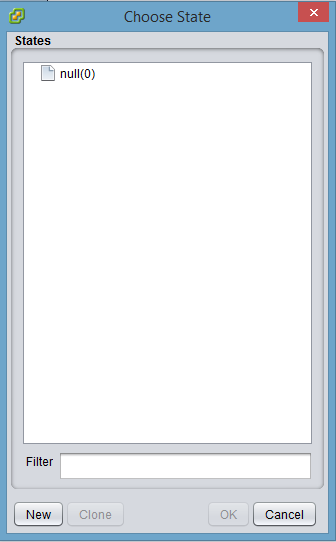
Now, let us follow similar steps and define states. You can define multiple states per test. In our example, the states to be defined are:

* group\_id = 0
* user\_id = 0
* Permissions 644
  + uexec = 0
  + gwrite = 0
  + gexec = 0
  + owrite = 0
  + oexec = 0

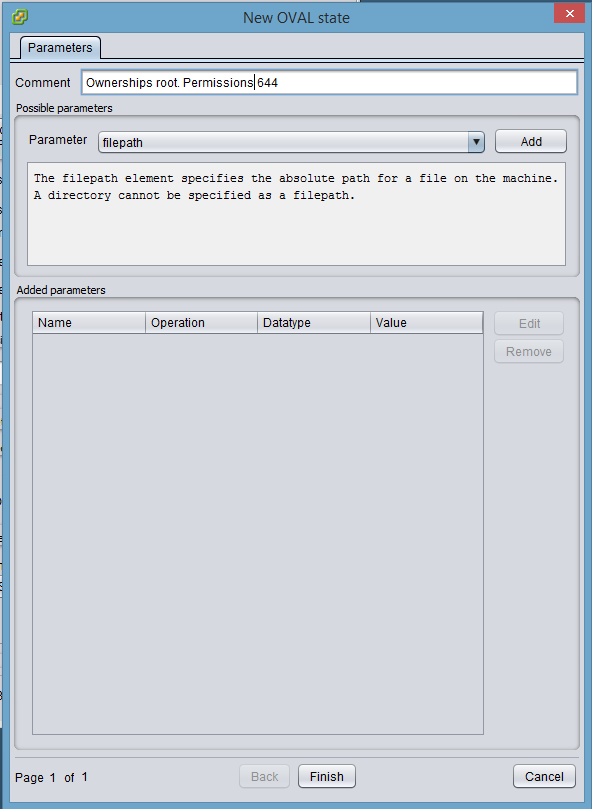
In the state definition area, click on **Add**.



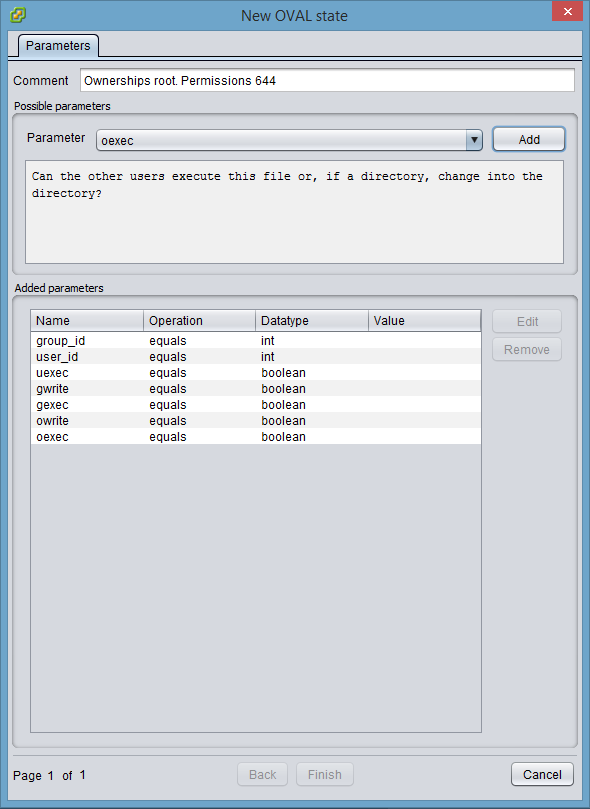
The state wizard opens up. Click **New**.



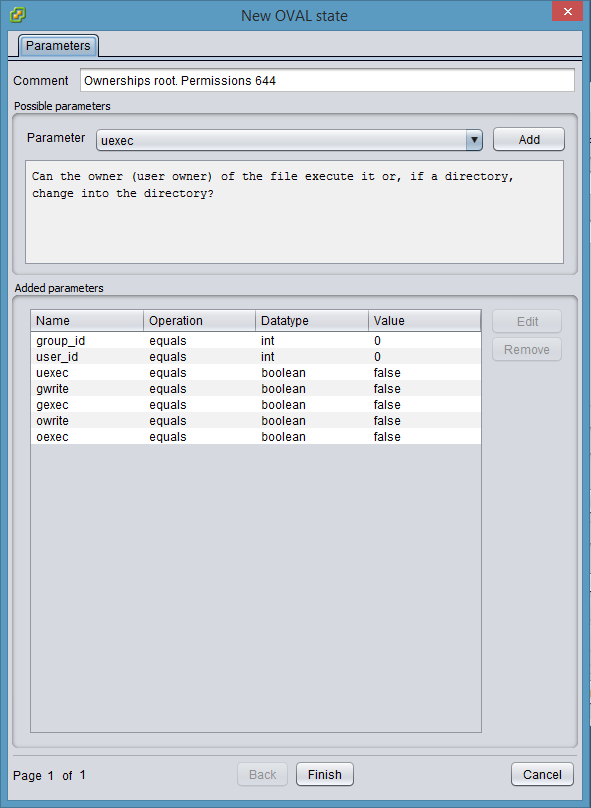
Define a new state. First, give a **Comment**.



Then, add all the state parameters needed from the parameter drop down list, one at a time.

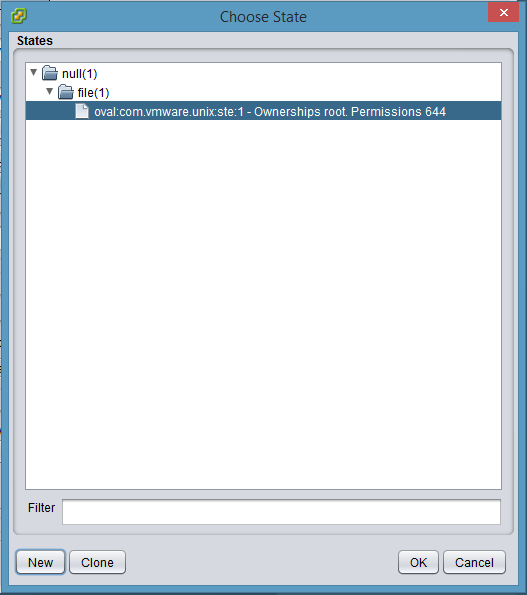


Then select each parameter and click **Edit** to set the desired value.



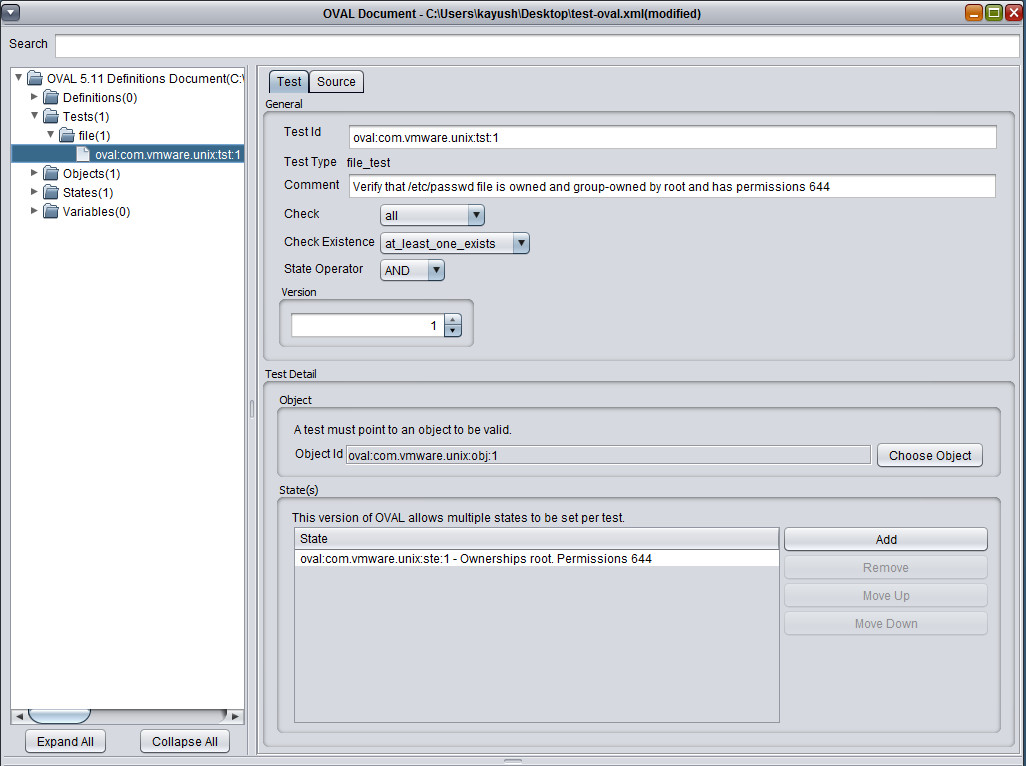
Click **Finish**.

Choose the state you just created.



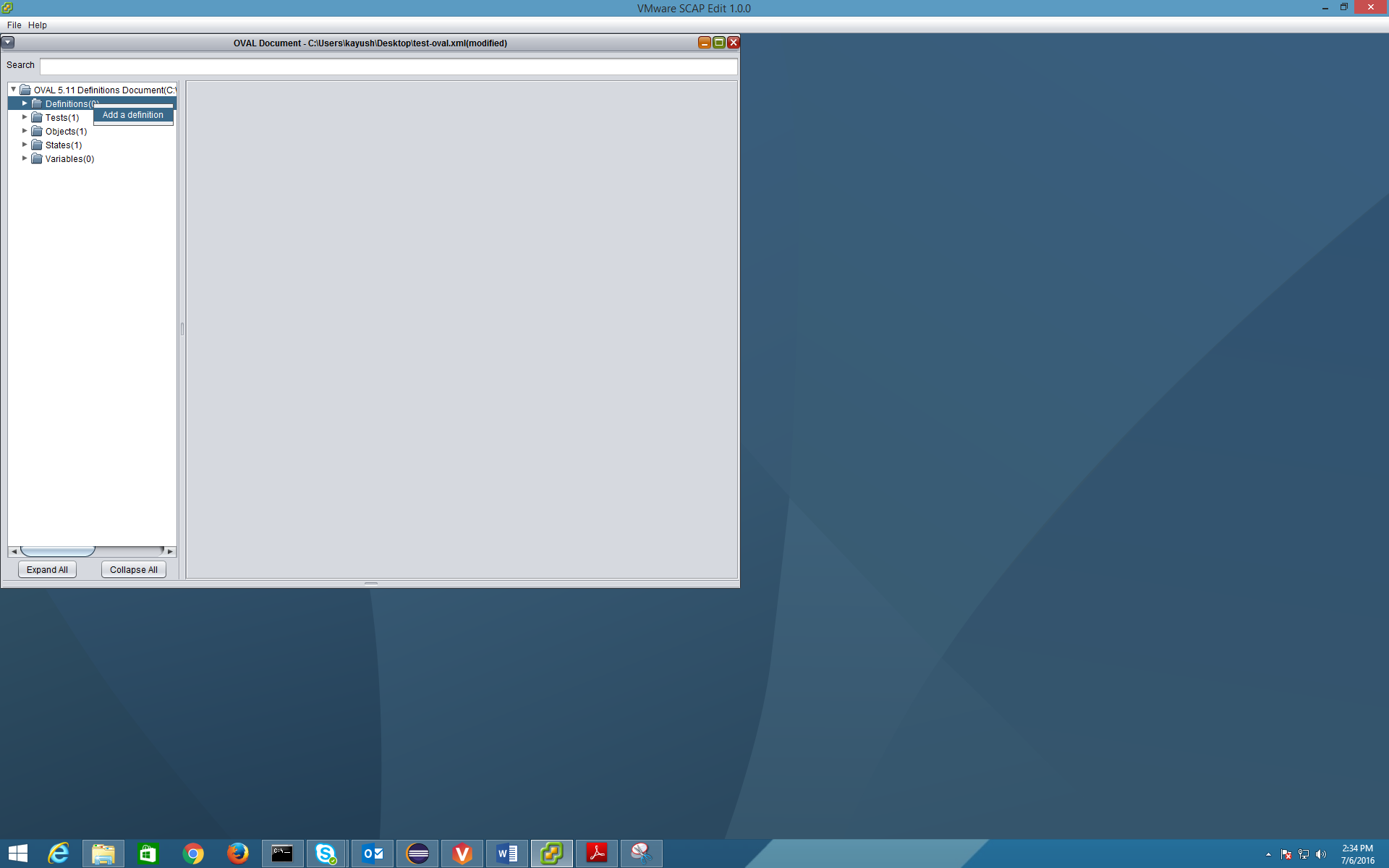
Click **OK**.

Now, the test is complete. We defined all the test elements.

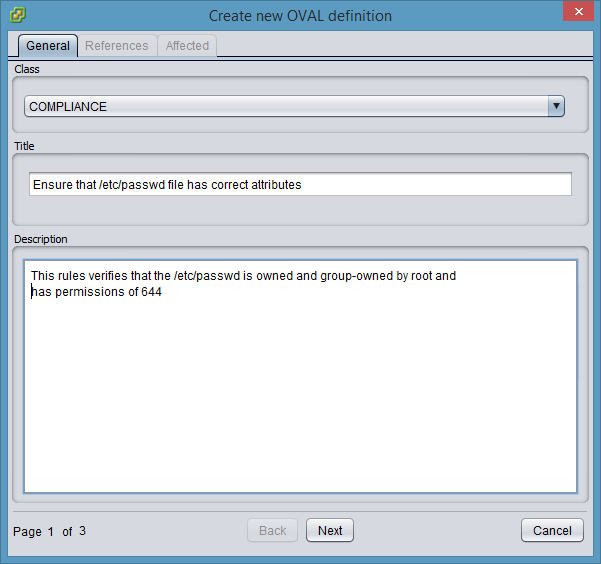


The next step is to link the test to an OVAL definition.

Right click on Definitions and select **Add a definition**.

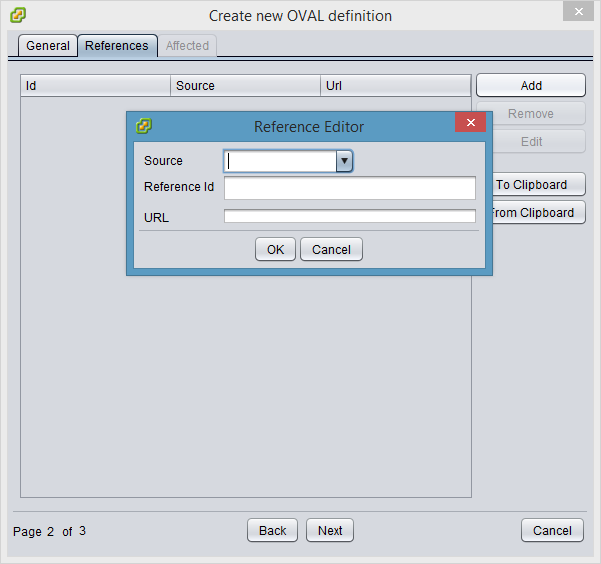


This would open OVAL definition wizard. In this wizard, choose the definition **Class** from the drop down and provide a suitable **Title** and **Description** for the definition.



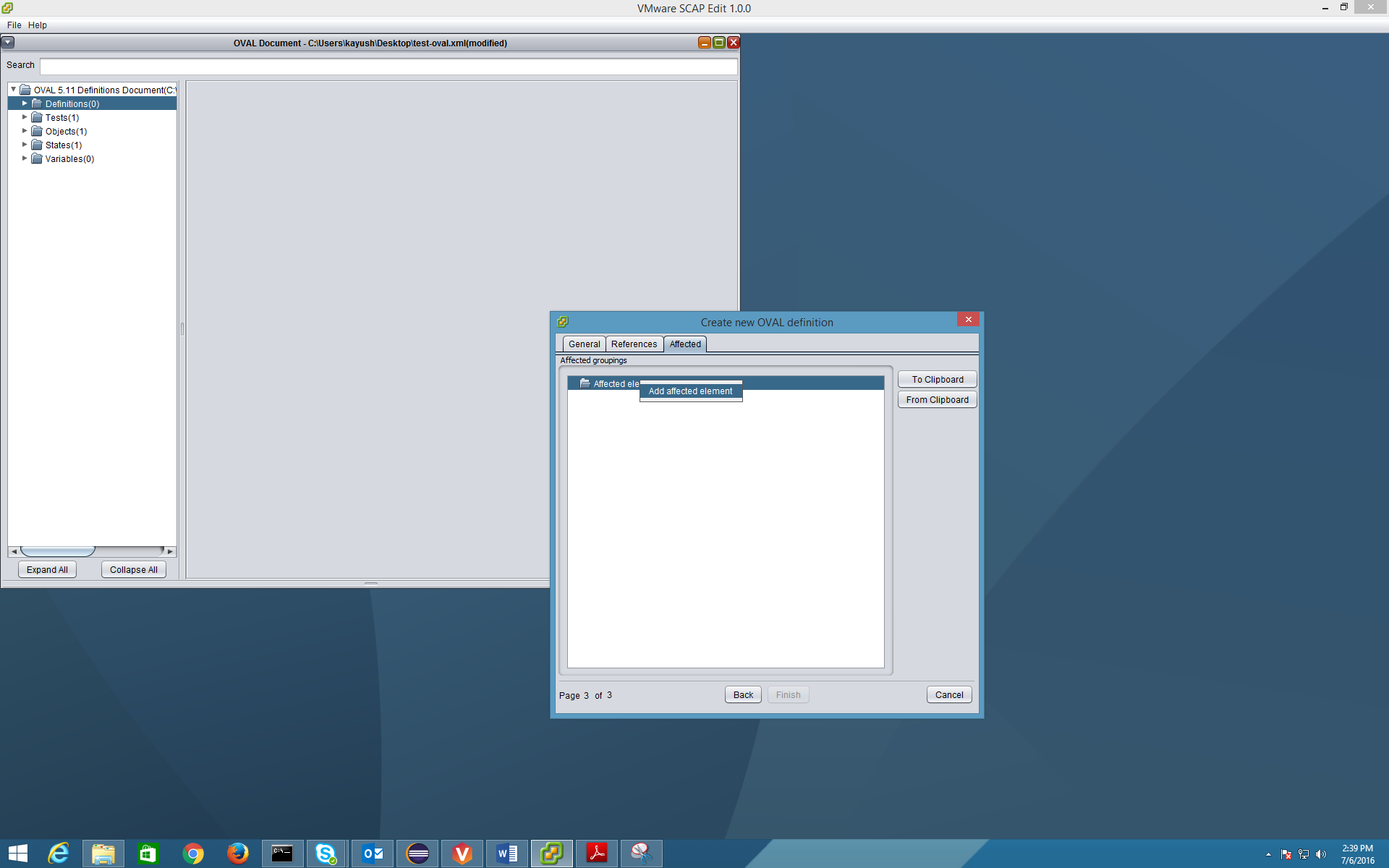
Click **Next**.

Add references if you want to. This is optional.

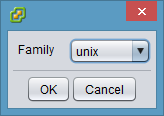


Click **Next**.

Add affected platforms and products. Right click on **Affected elements** and select **Add affected element**.

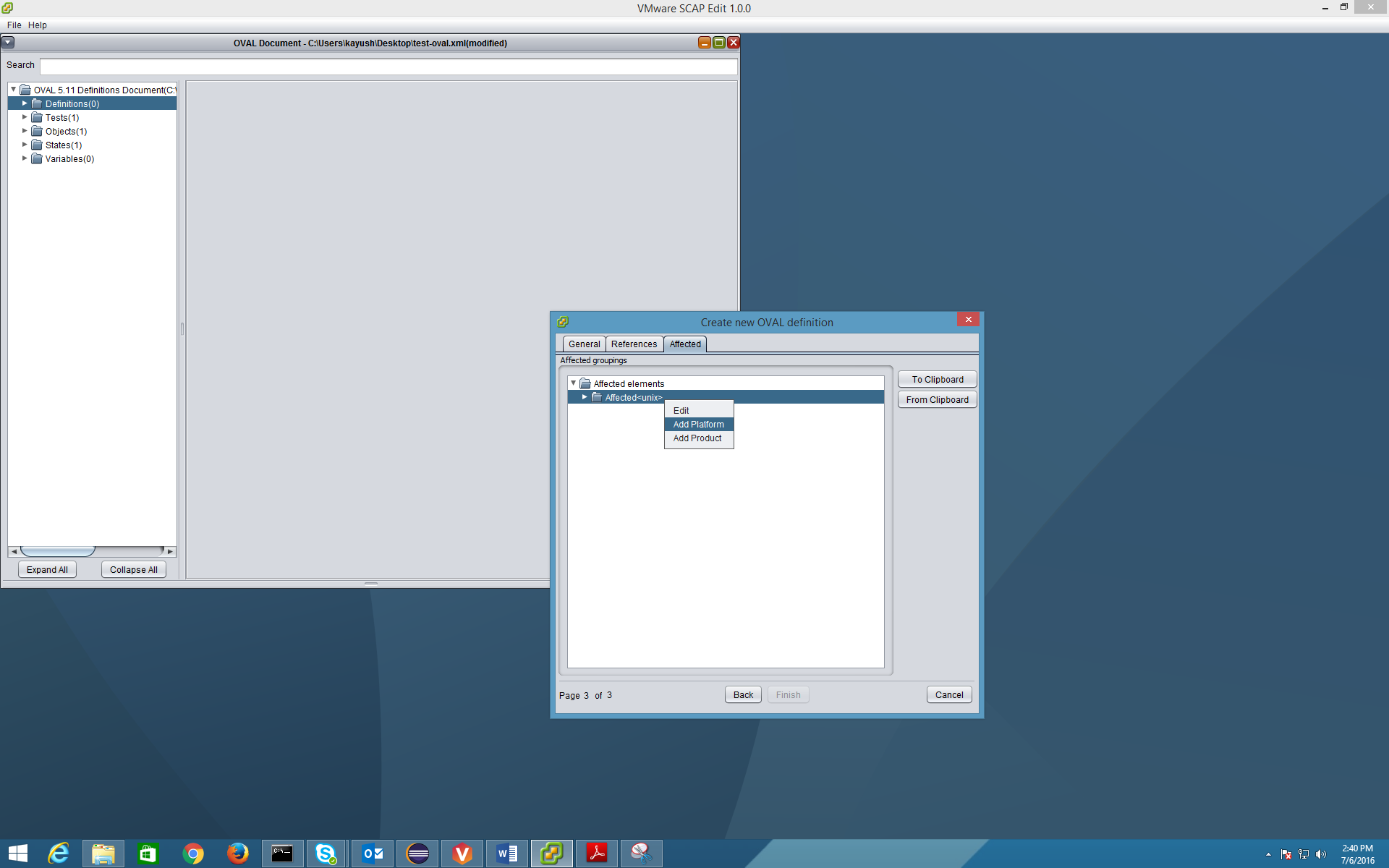


In our example, we will choose Unix.

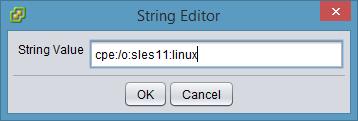


Click **OK**.

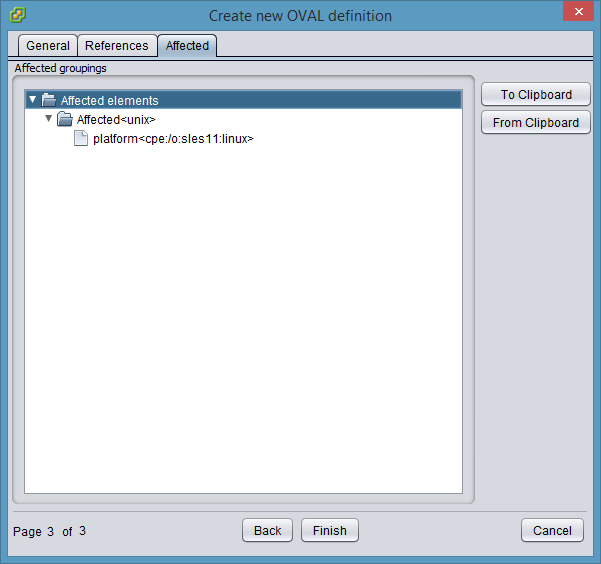
Then, define a specific platform or product. Right click on **Affected<unix>** and choose the target as needed.



For our example, let us give SLES 11 as the target platform. An entry for it will look like below.

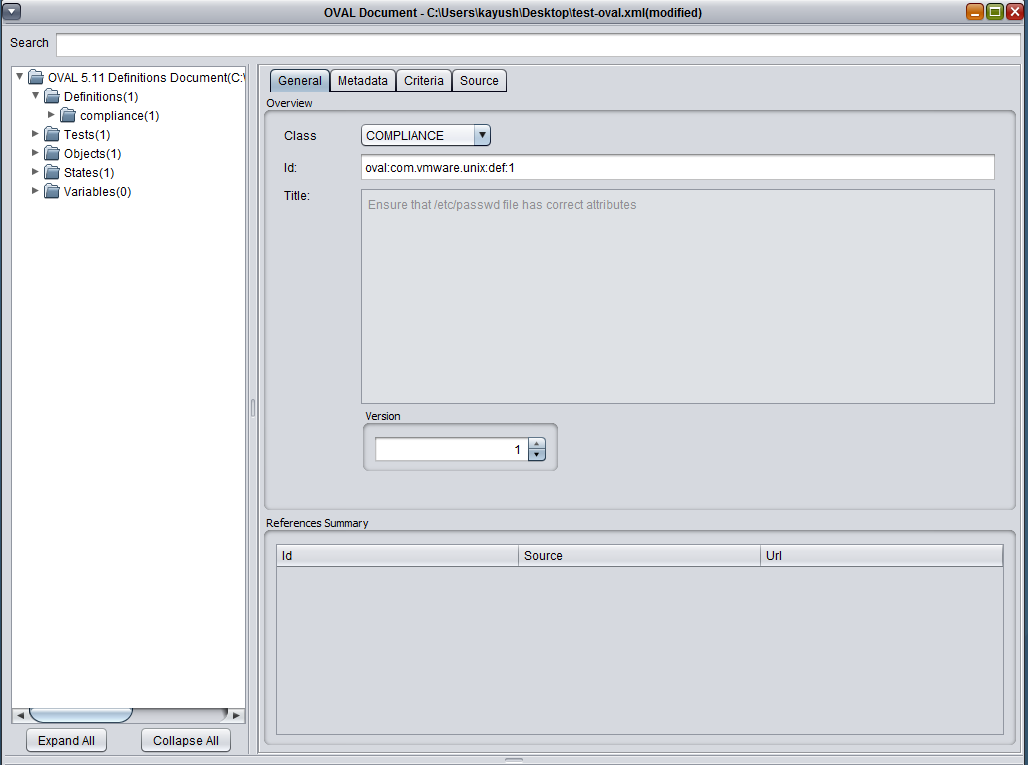


Click **OK**.

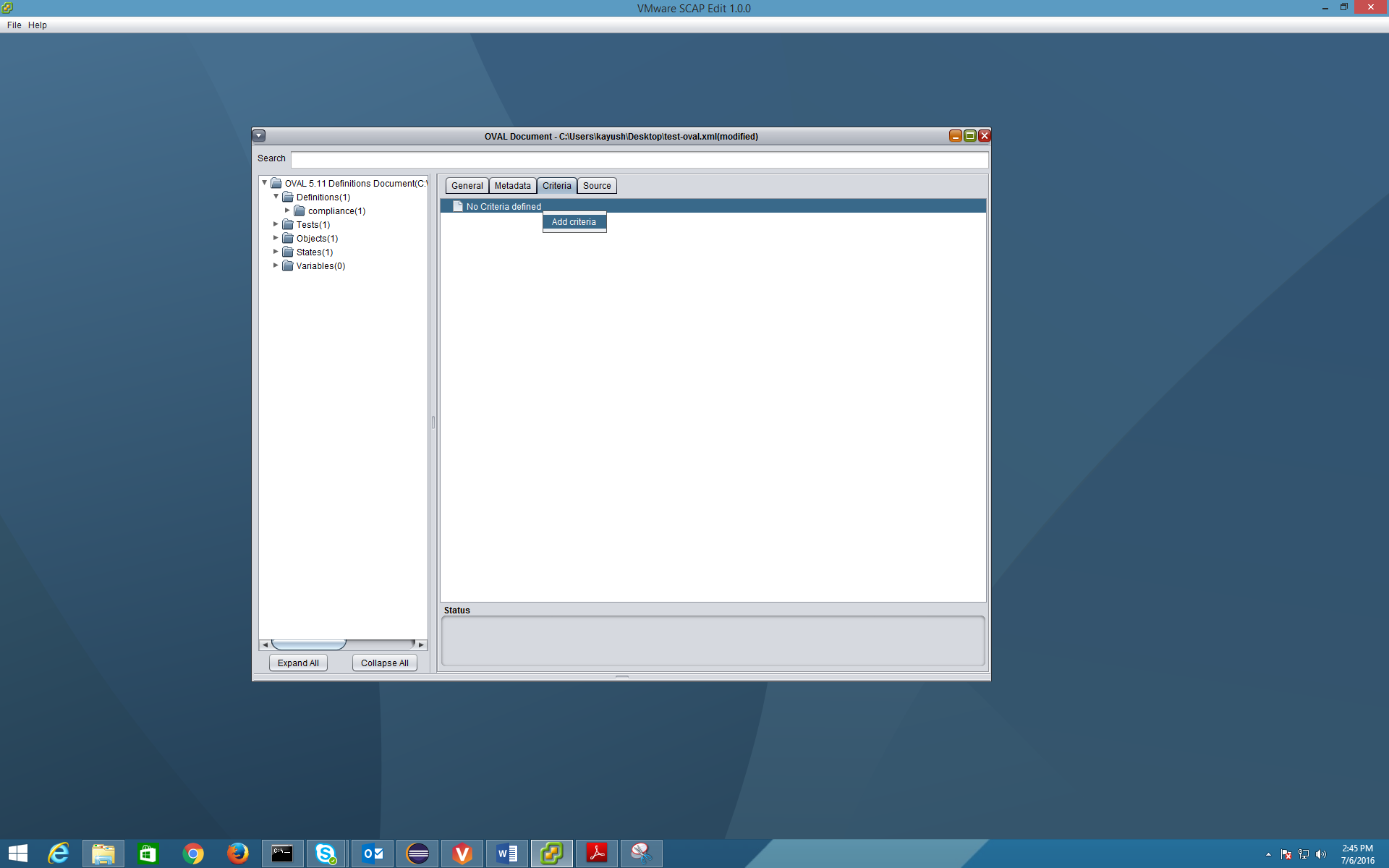


Click **Finish**.

This adds a new OVAL definition. Now, we will link this definition with a test. Definitions can have one or more tests. In our example, we will associate this definition with the test we created in earlier steps.

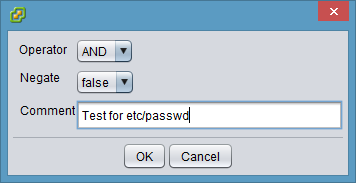


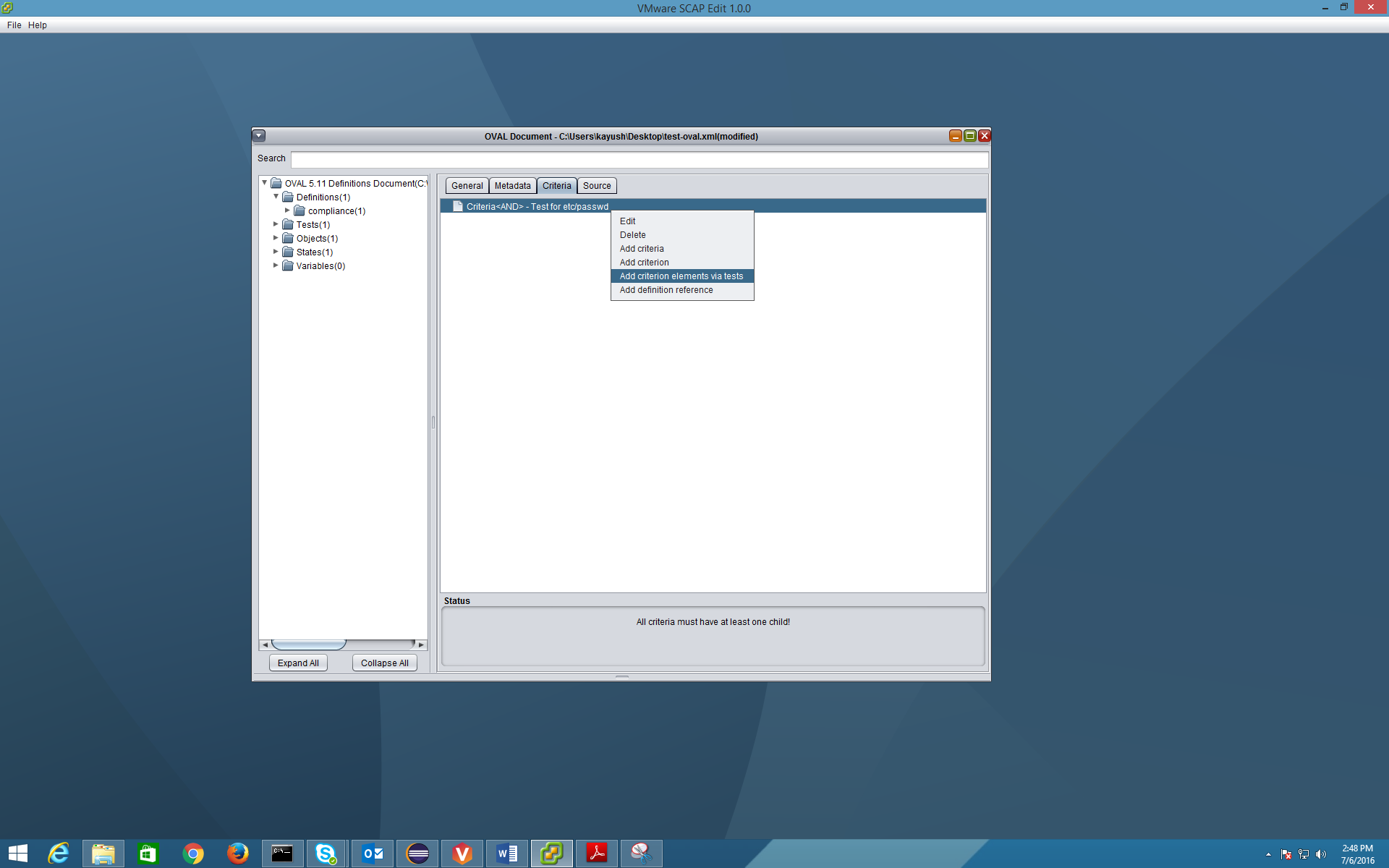
Click on **Criteria** tab.



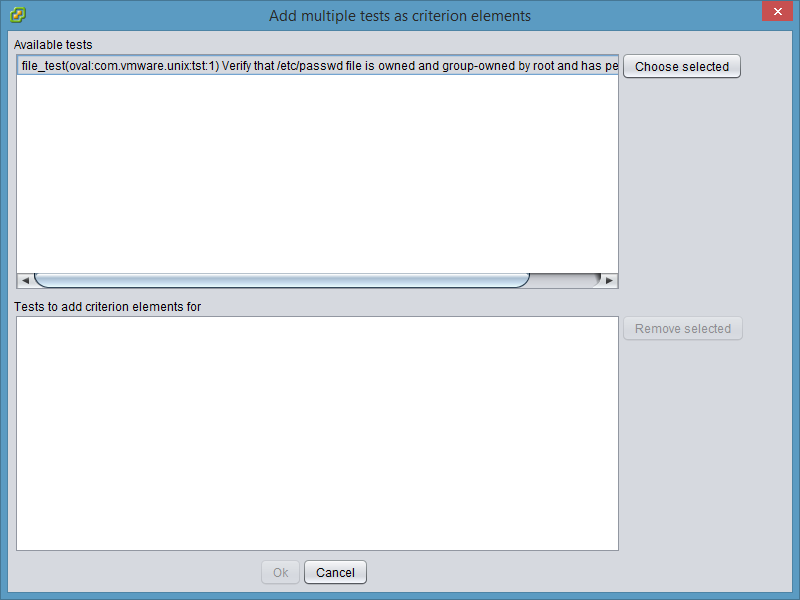
Right click on **No Criteria defined** and then choose **Add criteria**.

Choose criteria operators (defaults are preset) and provide a **Comment** and then click **OK**.





Now right click on **Criteria** to choose **Add criterion elements via tests**.

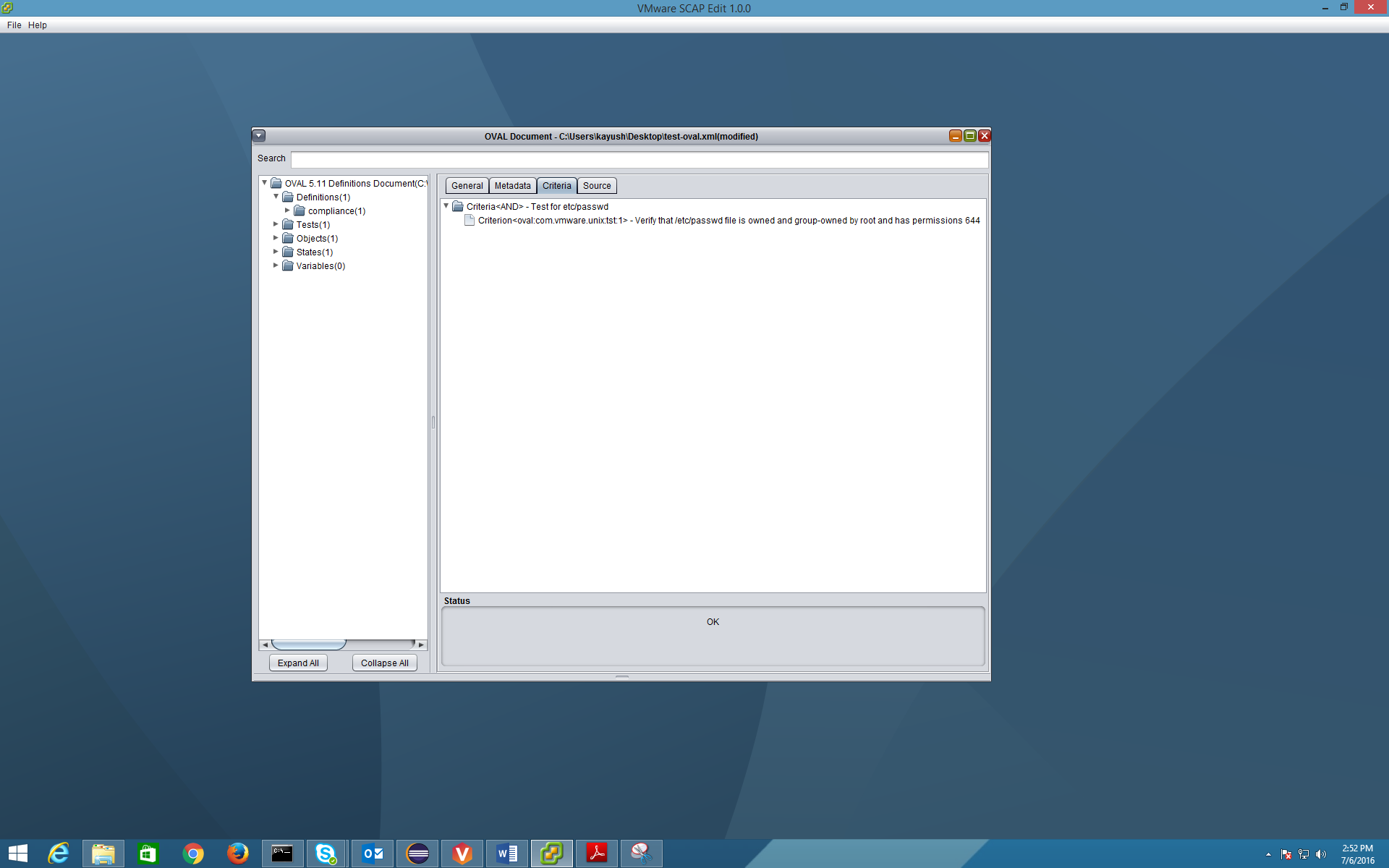


Select the desired **Test** and click on **Choose selected**.



Click **Ok**.

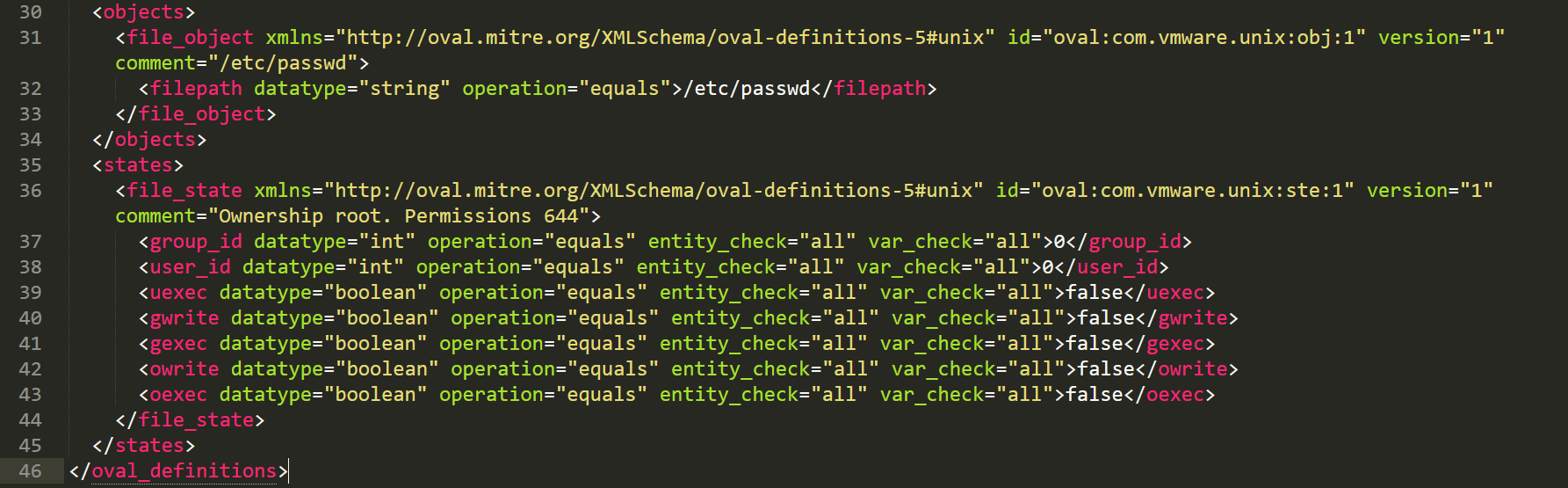
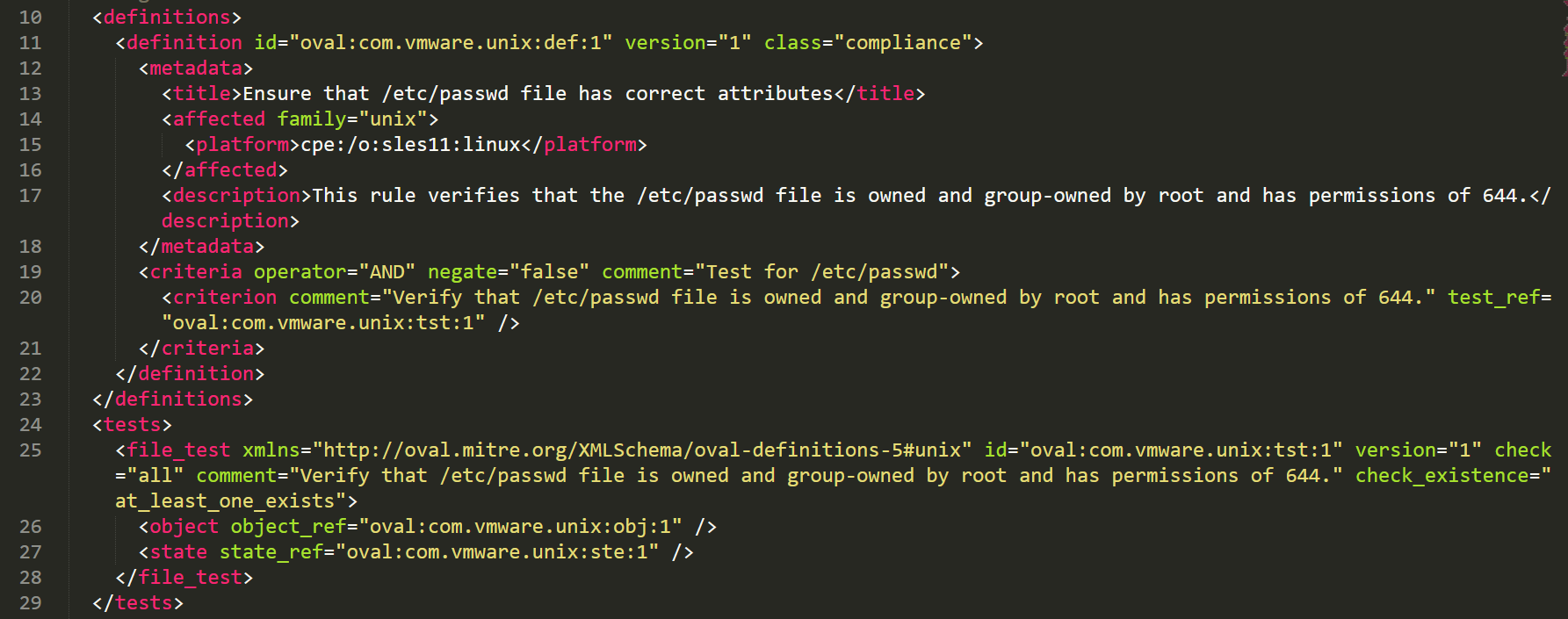
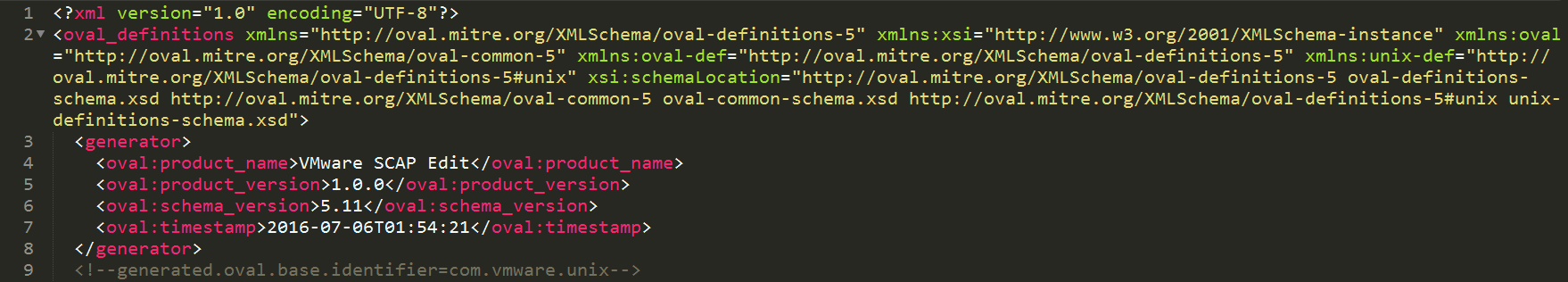
This will add the test to the definition.



Go to **File** menu and choose **Save**. Done!

This completes adding your first OVAL definition and the required elements.

The generated XML file looks like below:



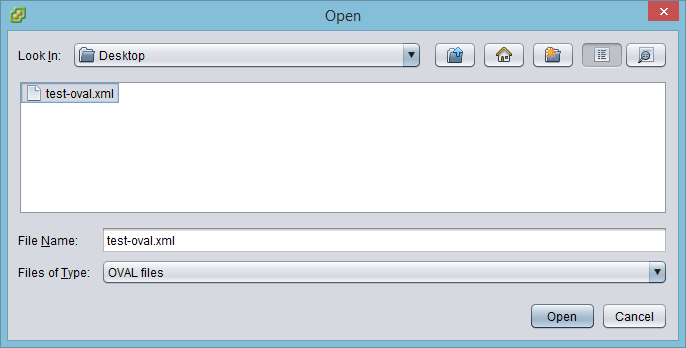
Now, you can run this OVAL definition using a tool such as [VMware STIG compliance tool](https://blogs.vmware.com/security/2016/05/vmware-releases-stig-compliance-app-for-free.html) on the target platform.

### Opening an existing OVAL file

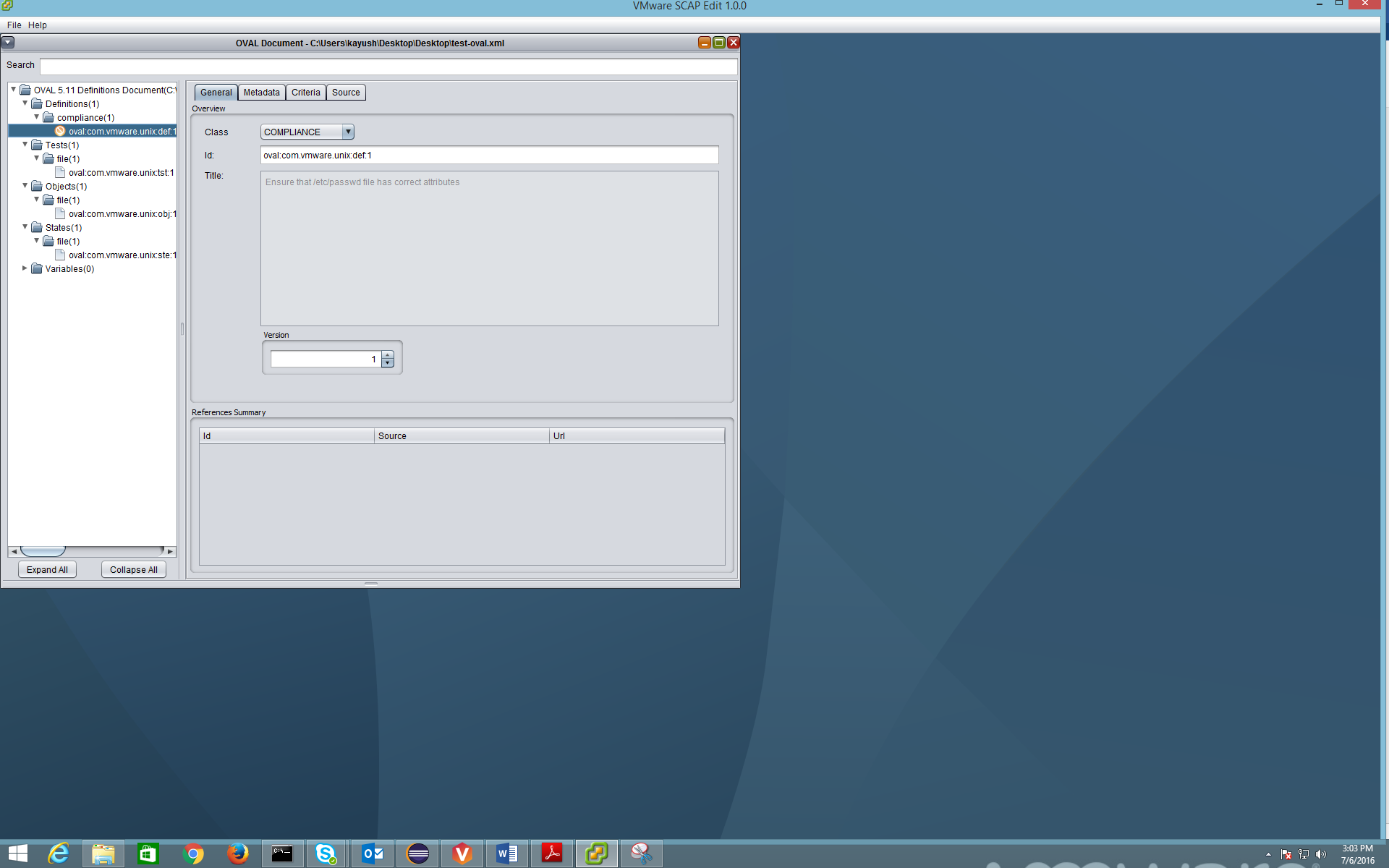
To open an existing OVAL file, go to **File 🡪 Open OVAL**.



Browse the file, select it and click **Open**.



This would open up the OVAL file that we just created.



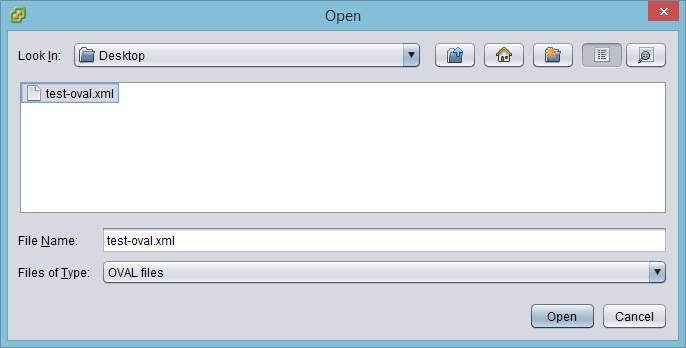
You may then add, edit or remove any OVAL elements to this file as desired. Do not forget to **Save** the file each time you have made changes to it that should be persisted.

### Creating XCCDF file from an existing OVAL file

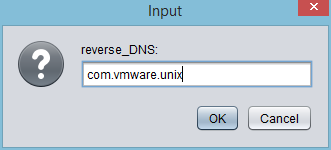
Using the tool, you can directly create a minimal XCCDF 1.2 file from an existing OVAL file. To do this, just launch the tool and click on **File 🡪 XCCDF from OVAL**.



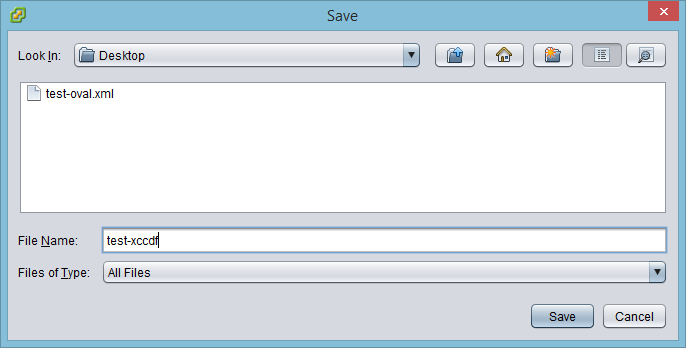
Browse the file, select it and click **Open**.



Provide the desired reverse DNS string and click **OK**.



Provide desired XCCDF file name and click **Save**.

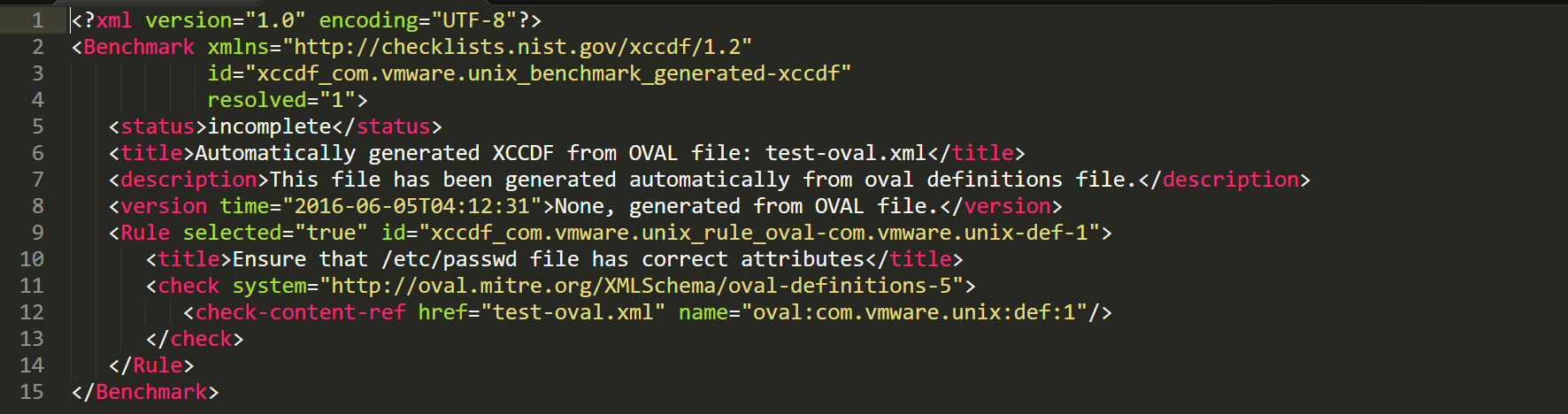


This would create a XCCDF 1.2 file from chosen OVAL file.



Click **OK**.

The generated XCCDF XML file looks like below:



Modify the XML tags such as status, title, description and version and then your XCCDF file is good to go. Optionally, you can manually create XCCDF profiles and groups, if required.

You can then do XCCDF assessments or use the XCCDF to create SCAP source data stream, if required.

# References

Below are some useful references.

OVAL core definition schema –<https://oval.mitre.org/language/version5.11/ovaldefinition/documentation/oval-definitions-schema.html>

OVAL independent definition schema –<http://oval.mitre.org/language/version5.11/ovaldefinition/documentation/independent-definitions-schema.html>

OVAL Linux definition schema –<http://oval.mitre.org/language/version5.11/ovaldefinition/documentation/linux-definitions-schema.html>

OVAL Unix definition schema –<http://oval.mitre.org/language/version5.11/ovaldefinition/documentation/unix-definitions-schema.html>

OVAL Windows definition schema – <http://oval.mitre.org/language/version5.11/ovaldefinition/documentation/windows-definitions-schema.html>

VMware STIG compliance tool –

<https://blogs.vmware.com/security/2016/05/vmware-releases-stig-compliance-app-for-free.html>