The Maven Minifilter provides runtime security for Windows 10 and Windows Server 2016 hosted machines. The service is provided in two parts, a driver and a user application. The driver is a minifilter as described in the link below.

<https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/file-system-minifilter-drivers>

Installation:

Copy the installation files to a local folder. The target must be configured to accept test signed drivers.

1. Open a command shell in with Administrator privileges and execute the following command.  
   > bcdedit -set TESTSIGNING ON
2. Reboot the board to allow the change to take effect. Right-click on the driver installation file, mavenMinifilter.inf and install.
3. Reboot the board.

The user application can be placed anywhere that is convenient. It must run under an account with Administrator privilege.

The user app may be autoloaded when logging into the admin account. To autostart the App, open the Task Scheduler and add an entry for a Basic Task, Run Only When User Is Logged On, and Run with Highest Privileges.

The driver hooks the kernel in the file create path. This is called whenever opening a file, either preexisting or new. It also hooks in the process map path when the kernel attempts to map a new process image. The driver has three modes of operation,

* Idle – The driver is loaded but not monitoring kernel events.
* Learn – The driver monitors kernel events and builds on internal configuration table.
* Filter – The driver monitors kernel events and checks that file open and map requests are consistent with the permissions saved in the internal configuration table.

The user app, mavenUser.exe, runs as a shell application. It connects to the kernel driver using message pipes. It manages state changes within the driver and allows uploading and downloading driver configuration files at runtime. The user application accepts one-character commands. Available commands are displayed at startup in a help screen.

It is intended for the driver to load at boot time, filtering kernel events as early as possible in the boot sequence to protect the system from threats. To support this mechanism, the driver must load a valid and complete configuration file to be used to populate the internal table. This file is loaded at the location below.

C:\DriverTest\maven.csv

NOTE: The folder \DriverTest\ is a protected folder and should be used to place the maven.csv configuration file. Access to this folder is prevented by the filter when it is enabled.

Testing has shown that several hundred events must be filtered during this sequence. Populating the table by hand would be impractical. The learn mechanism is used to initially populate the driver configuration table. If the driver is unable to find the configuration file at the predefined location, it transitions to Learn mode and begins populating its internal table automatically. The user can then upload the table that is learned and save it off in the appropriate folder with the correct name. The driver will then load this table automatically at startup.

Testing has also shown that the Windows startup sequence is non-deterministic. If the boot sequence includes an event that is not preconfigured in the configuration file, the boot sequence will fail, leaving the board in an unusable condition. It is necessary to iteratively boot the board and allow the configuration file to grow until it stops growing. To accomplish this, the user must hand-edit the configuration file, putting an asterisk “\*” character in the first file position. The driver uses this as a key to recognize that during its initialization sequence it should transition to Learn mode instead of Filter mode. The user must repeat the following steps until the resulting configuration file stops growing. This may take a dozen or more iterations.

1. Boot the board
2. Wait until booting completes
3. Put the driver in idle

>i

1. Save the learned configuration off to the desktop as "maven.csv". The location of the mavenUser app is the default folder.  
    >u  
    enter filename: maven.csv
2. Edit this file with notepad and add a star character to the beginning of the file. This is used by the driver to indicate that learning is not complete and that on the next boot, it should read the configuration file and add to it in learn mode. This is crude, but it works for a prototype.  
   E.G. \*000000,\Device\HarddiskVolume1\Windows\System32\drivers\null.sys,1,0
3. Save the file off in the configuration directory.
4. Examine the file size and see if it has stopped growing. You should verify that the file size is consistent for at least three boots. If it is not yet complete, go back to step 1

NOTE: If you install this driver with an incomplete configuration file, the board may not boot unless safe mode is used. Deleting the offending configuration file will fix the problem. Safe mode may not be an option with a virtual machine however.

It is not necessary to have the driver find a configuration file. If none is found, the driver will still start but will not automatically begin filtering. You can play with the driver by deleting what it learns at startup using the reset command “r”, and then putting it back in learn. After teaching it some apps, put it in filter “f” to see how it behaves. If the board misbehaves, you can safely reboot. Learned apps are not automatically saved on the disk. You must explicitly upload the new configuration file and save it off as described above.