

AppLocker in Windows 10 Enterprise

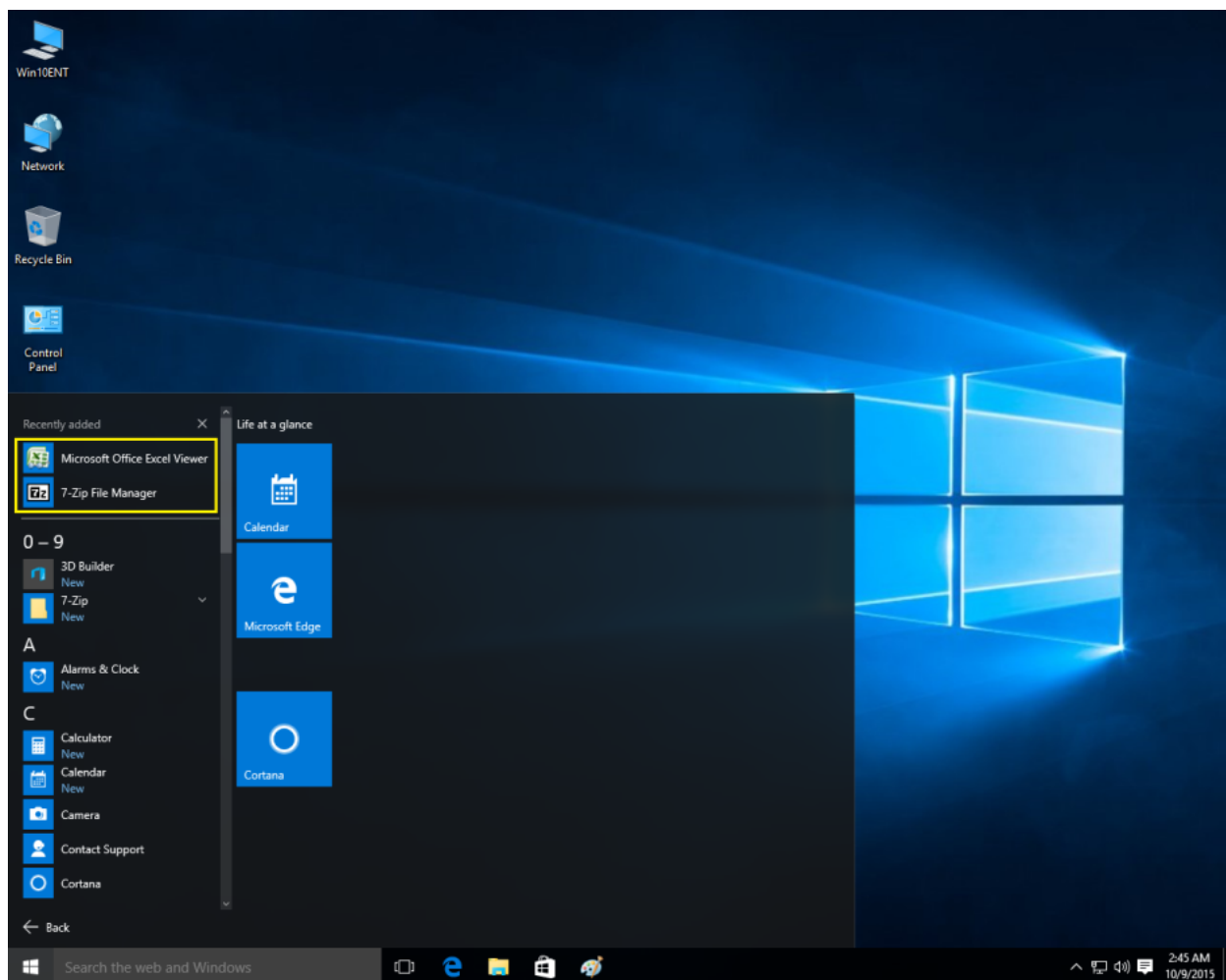
 michaelfirsov.wordpress.com/applocker-in-windows-10-enterprise

October 14, 2015

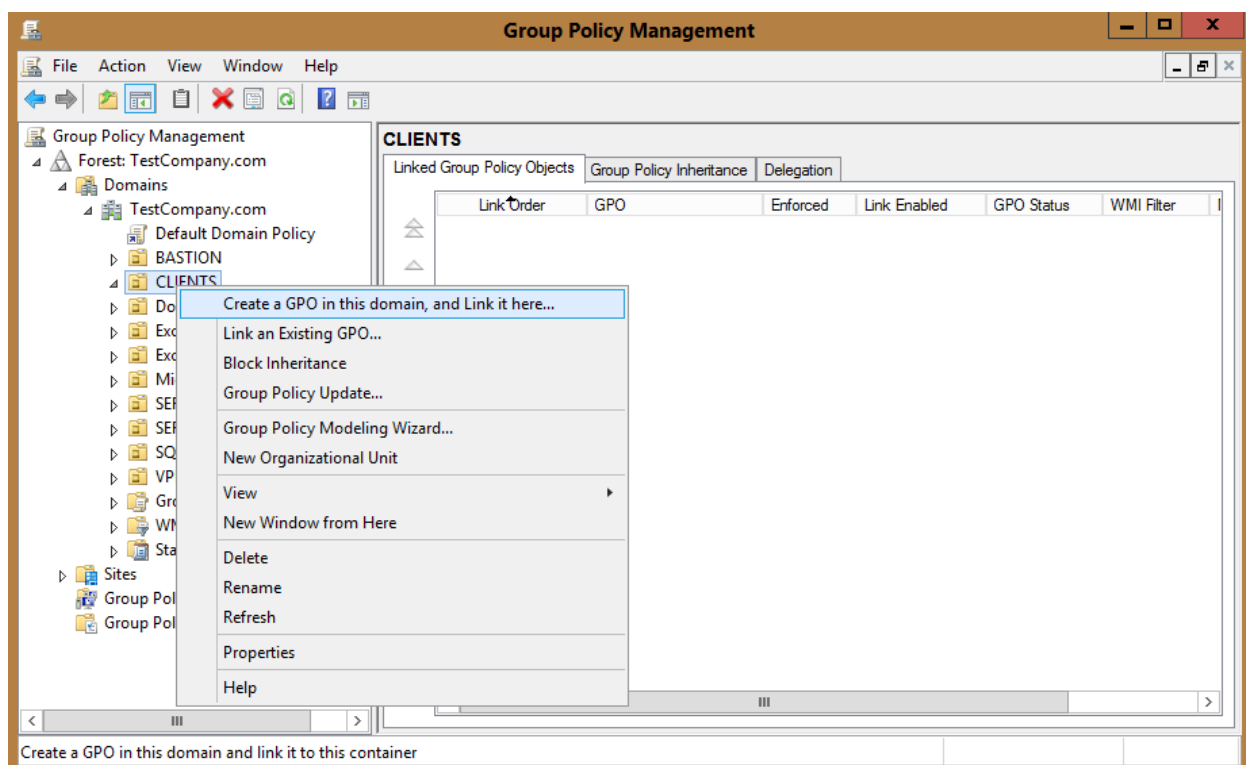
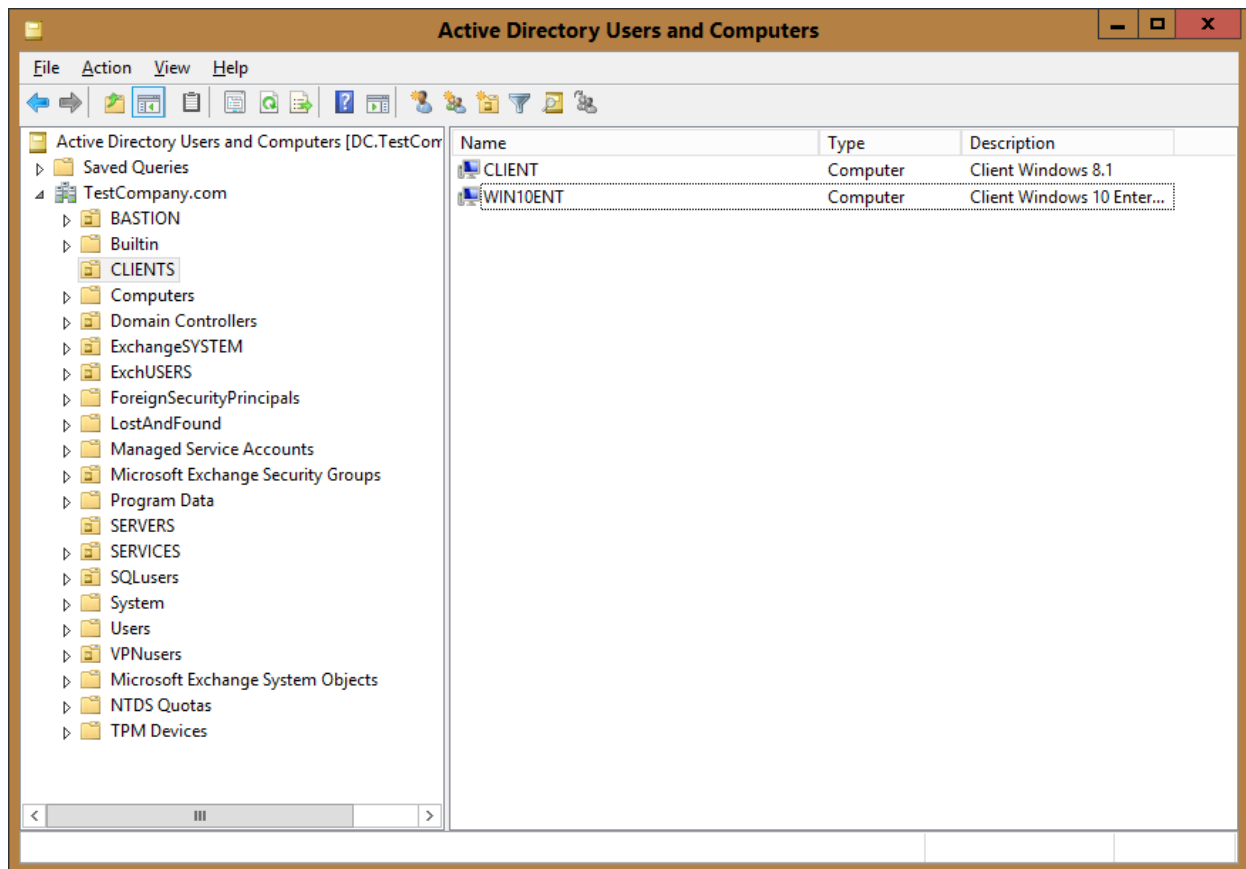
In this article I'd like to show how we can use **Windows AppLocker** in **Windows 10 Enterprise** to allow only a small subset of programs to run in an enterprise environment. As you already may know AppLocker rules function as an "allow" list meaning that you're allowed to run only those applications which have the corresponding allow rules in the AppLocker policy.

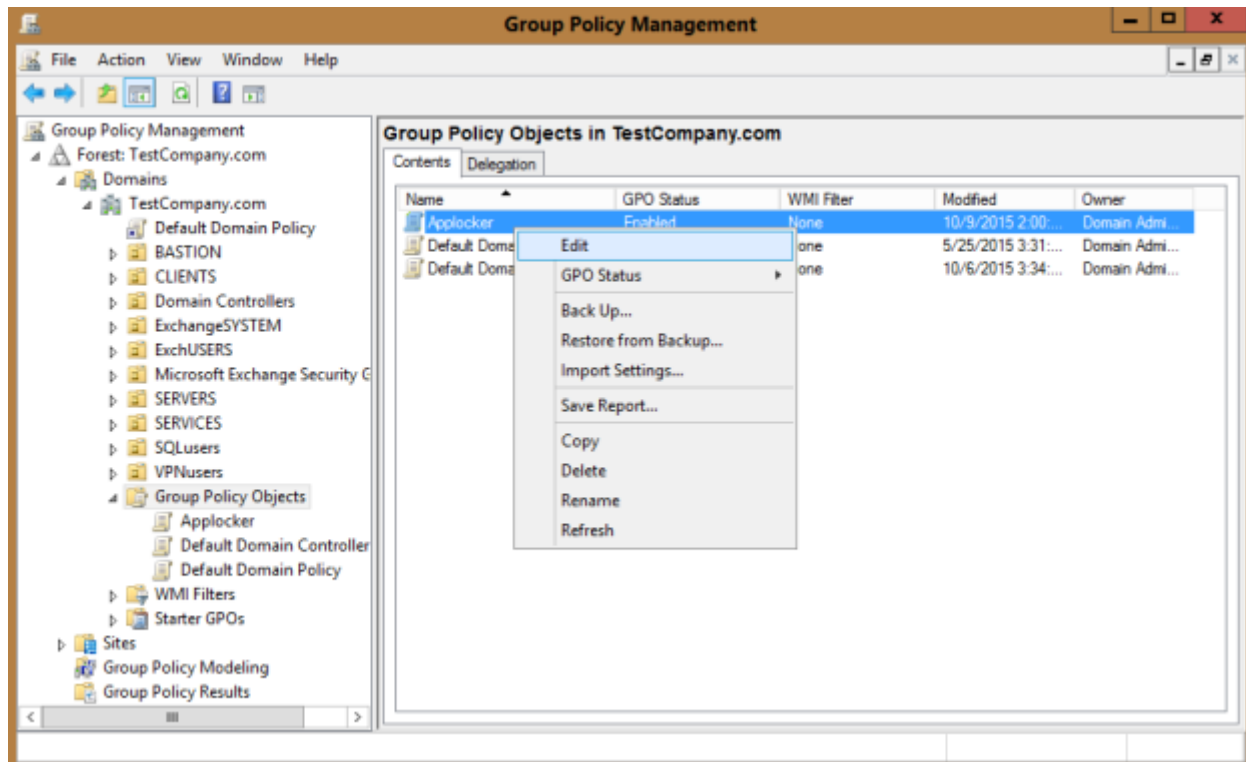
Suppose our goal is to restrict users to run only a single third-party application installed by an administrator, for example 7Zip. Theoretically we must use a sample PC with the needed applications installed for creating an Applocker policy locally and then exporting it to Active Directory GPO, but for the sake of this test I will create my Applocker policy using 7Zip installed on my DC.

To start with, let's take a look at my client computer – Win10Ent (Applocker policies may be applied only to enterprise OS versions!):

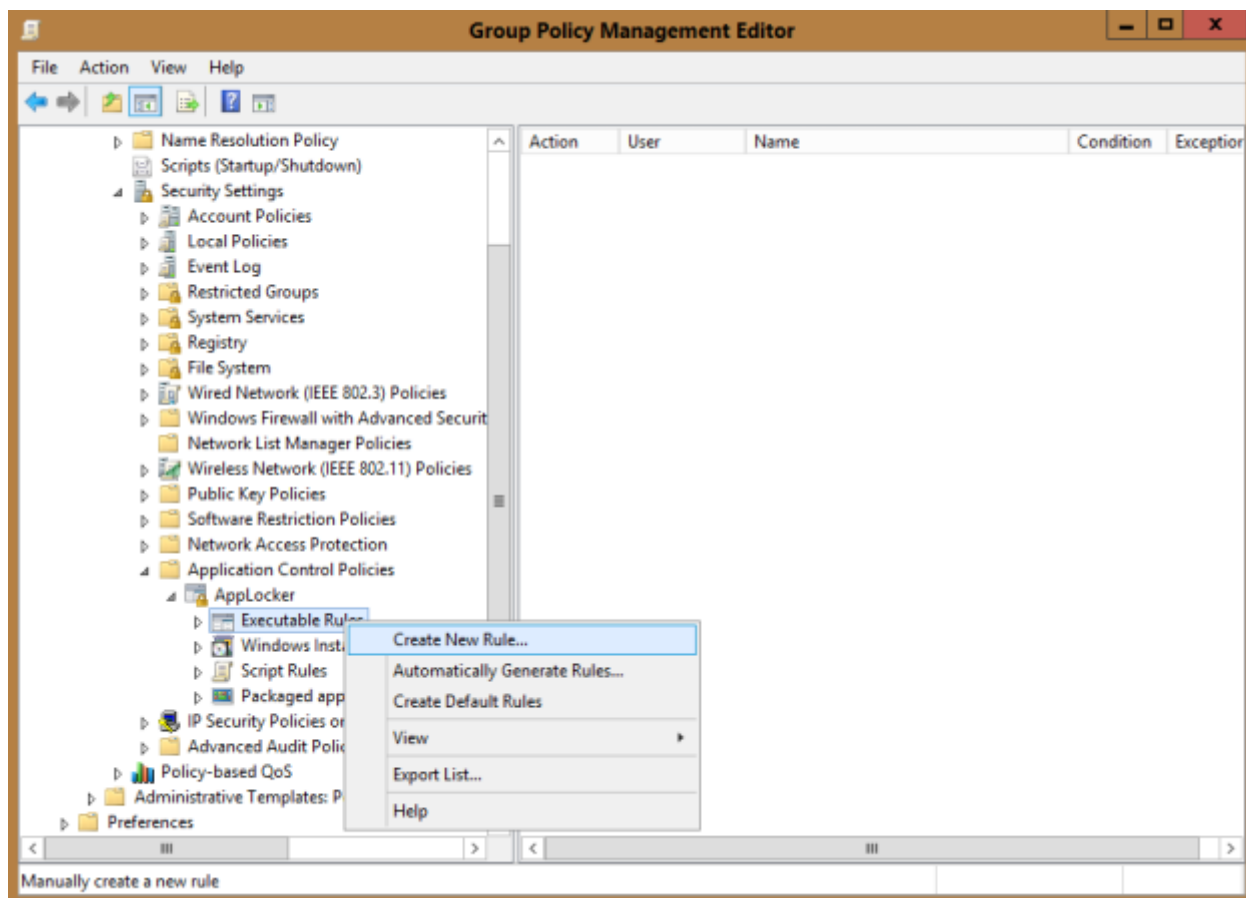


As we see there're two recently installed programs – 7Zip and MS Excel Viewer – I've installed them under the *TestCompany\ExAdmin* account. Now I want any other non-administrative users to run only one of these programs – 7Zip and NOT MS Excel Viewer. As I'd like to have the same policy for all of my clients I'll create a GPO in AD and deploy it for the **CLIENTS** OU:





Advertisements
Report this adPrivacy



Create Executable Rules

Before You Begin

Before You Begin

Permissions

Conditions

Publisher

Exceptions

Name

This wizard helps you create an AppLocker rule. A rule is based on file attributes, such as the file path or the software publisher contained in the file's digital signature.

Before continuing, confirm that the following steps are complete:

- Install the applications you want to create the rules for on this computer.
- Back up your existing rules.
- Review the AppLocker documentation.

To continue, click Next.

☐ Skip this page by default

< Previous


Next >

Create

Cancel

4/27

Create Executable Rules



Permissions

Before You Begin

Permissions

Conditions

Publisher

Exceptions

Name

Select the action to use and the user or group that this rule should apply to. An allow action permits affected files to run, while a deny action prevents affected files from running.

Action:

☒ Allow

☐ Deny

User or group:

TESTCOMPANY\Domain Users

Select...

[More about rule permissions](#)


< Previous

Next >

Create

Cancel

Create Executable Rules



Conditions

Before You Begin

Permissions

Conditions

Path

Exceptions

Name

Select the type of primary condition that you would like to create.

☐ Publisher

Select this option if the application you want to create the rule for is signed by the software publisher.

☒ Path

Create a rule for a specific file or folder path. If you select a folder, all files in the folder will be affected by the rule.

☐ File hash

Select this option if you want to create a rule for an application that is not signed.

[More about rule conditions](#)

< Previous

Next >

Create

Cancel

Create Executable Rules

Path

Before You Begin

Permissions

Conditions

Path

Exceptions

Name

Select the file or folder path that this rule should affect. If you specify a folder path, all files underneath that path will be affected by the rule.

Path:

%PROGRAMFILES%\7-Zip*

Browse Files...

Browse Folders...

[More about path rules and path variables](#)

< Previous

Next >

Create

Cancel

Create Executable Rules

Exceptions

Before You Begin

Permissions

Conditions

Path

Exceptions

Name

To add an exception, select the type of exception and then click Add. Exceptions are optional and allow you to exclude files that would normally be included in the rule. To continue configuring this rule without adding an exception, click Next.

Primary condition:
%PROGRAMFILES%\7-Zip*

Add exception:
Publisher

Exceptions:

| Exception | Type |
|-----------|------|
|-----------|------|

Add...
Edit
Remove


< Previous

Next >

Create

Cancel

AppLocker



The default rules are currently not in the rule list for this rule collection. When creating rules, it is recommended that you also create the default rules to ensure that important system files will be allowed to run.

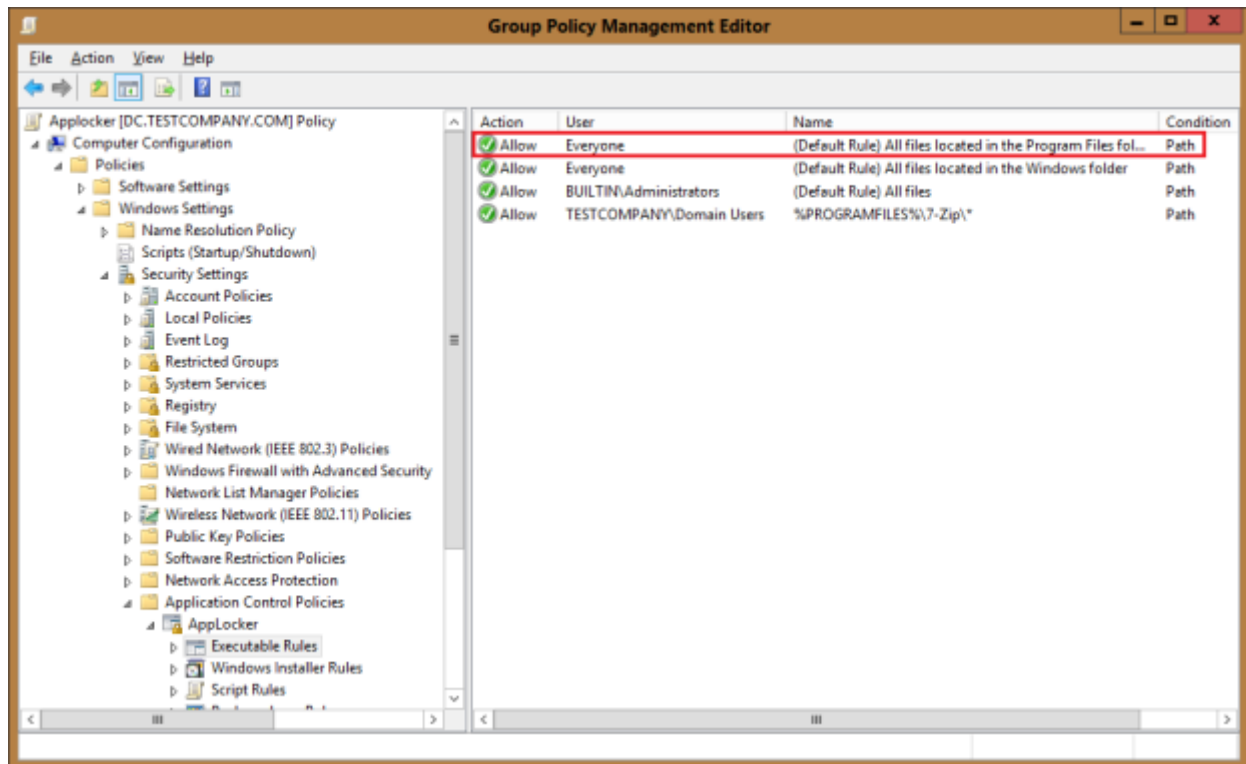
Do you want to create the default rules now?

Yes

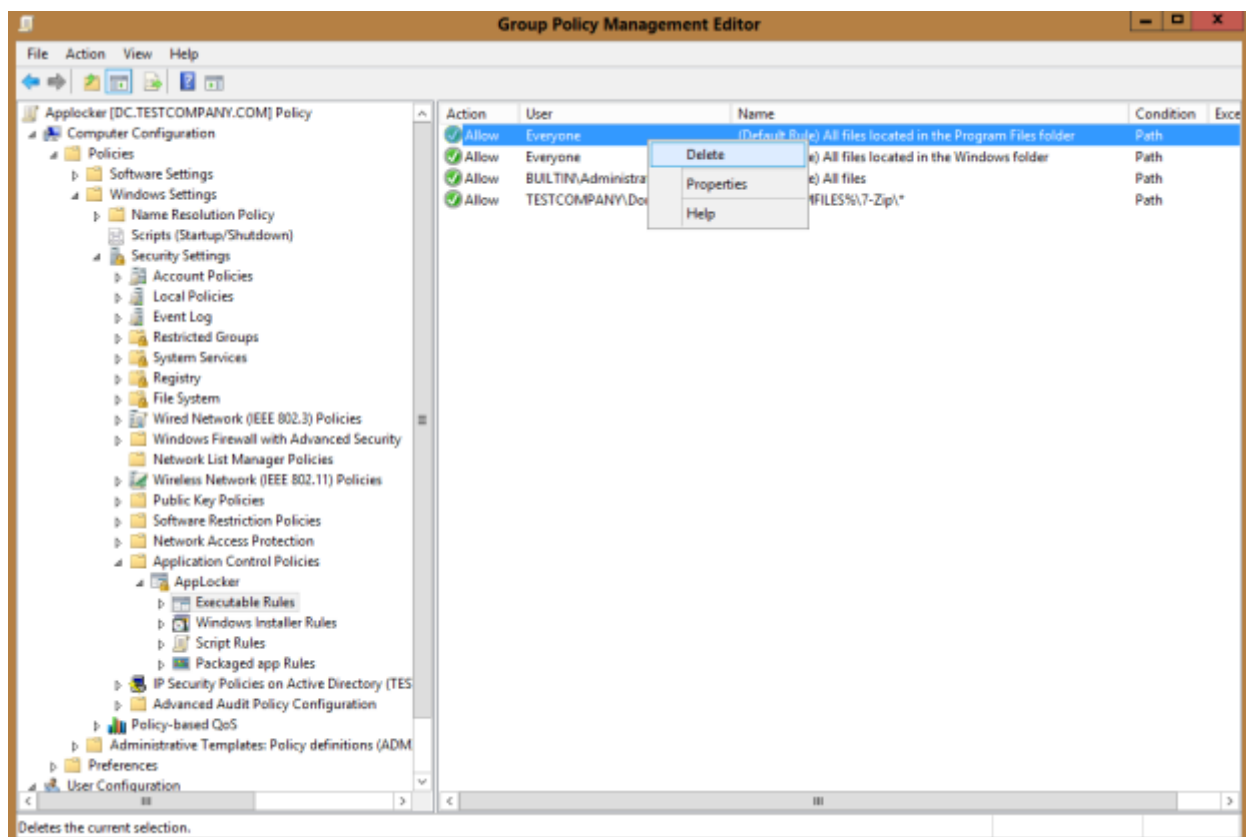
No

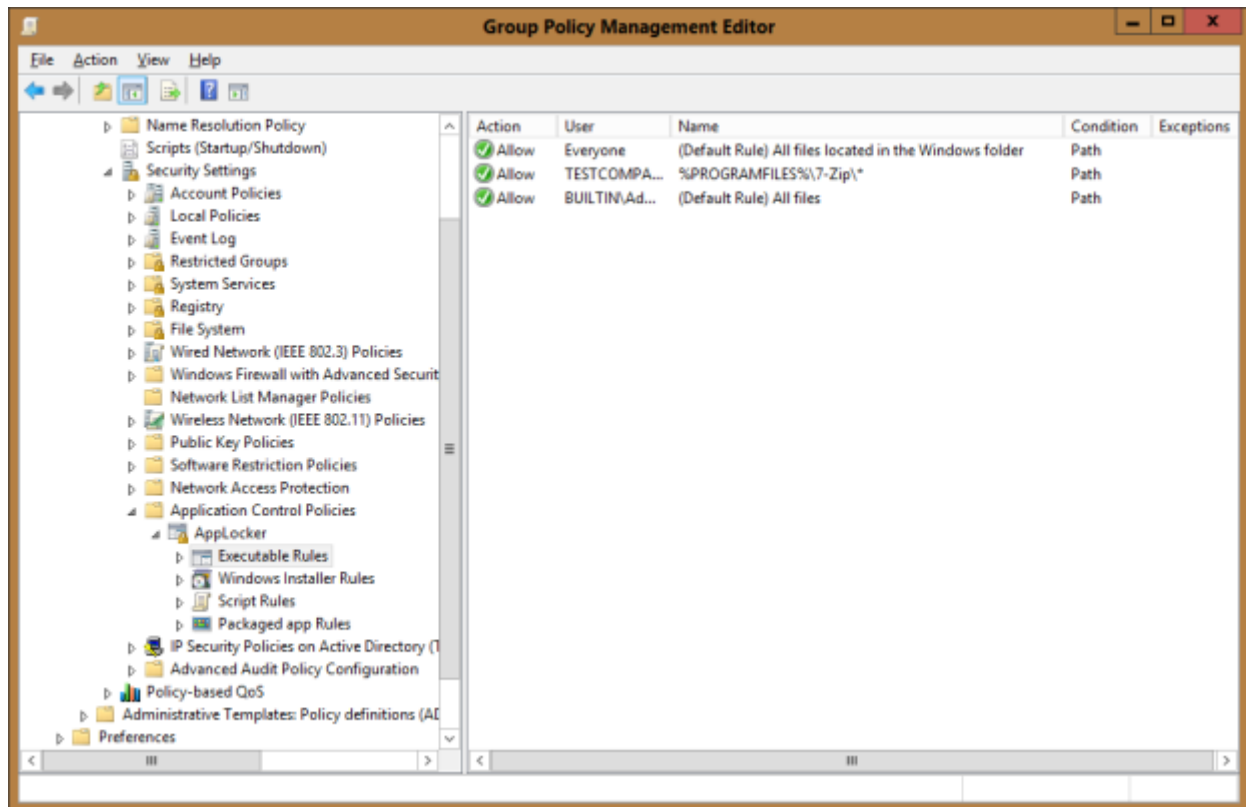
Advertisements
Report this adPrivacy

8/27

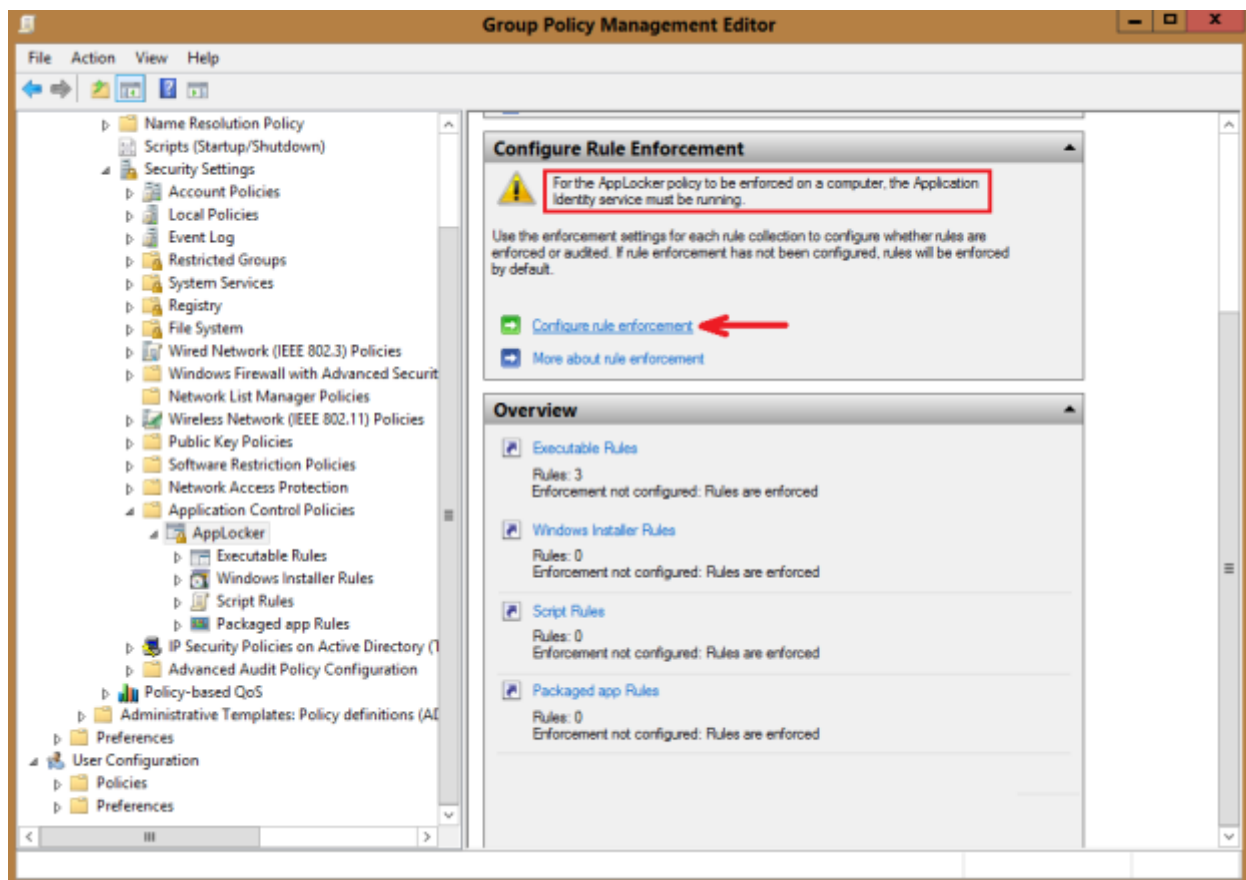


The first default rule that allows everyone to run programs located in the Program Files folder must be deleted – otherwise MS Excel Viewer will be implicitly allowed to run for all users.





Now we must enforce the rules:



AppLocker Properties

Enforcement

Advanced

Specify whether AppLocker rules are enforced for each rule collection.

Executable rules:

☒ Configured

Enforce rules

Windows Installer rules:

☐ Configured

Enforce rules

Script rules:

☐ Configured

Enforce rules

Packaged app Rules:

☐ Configured

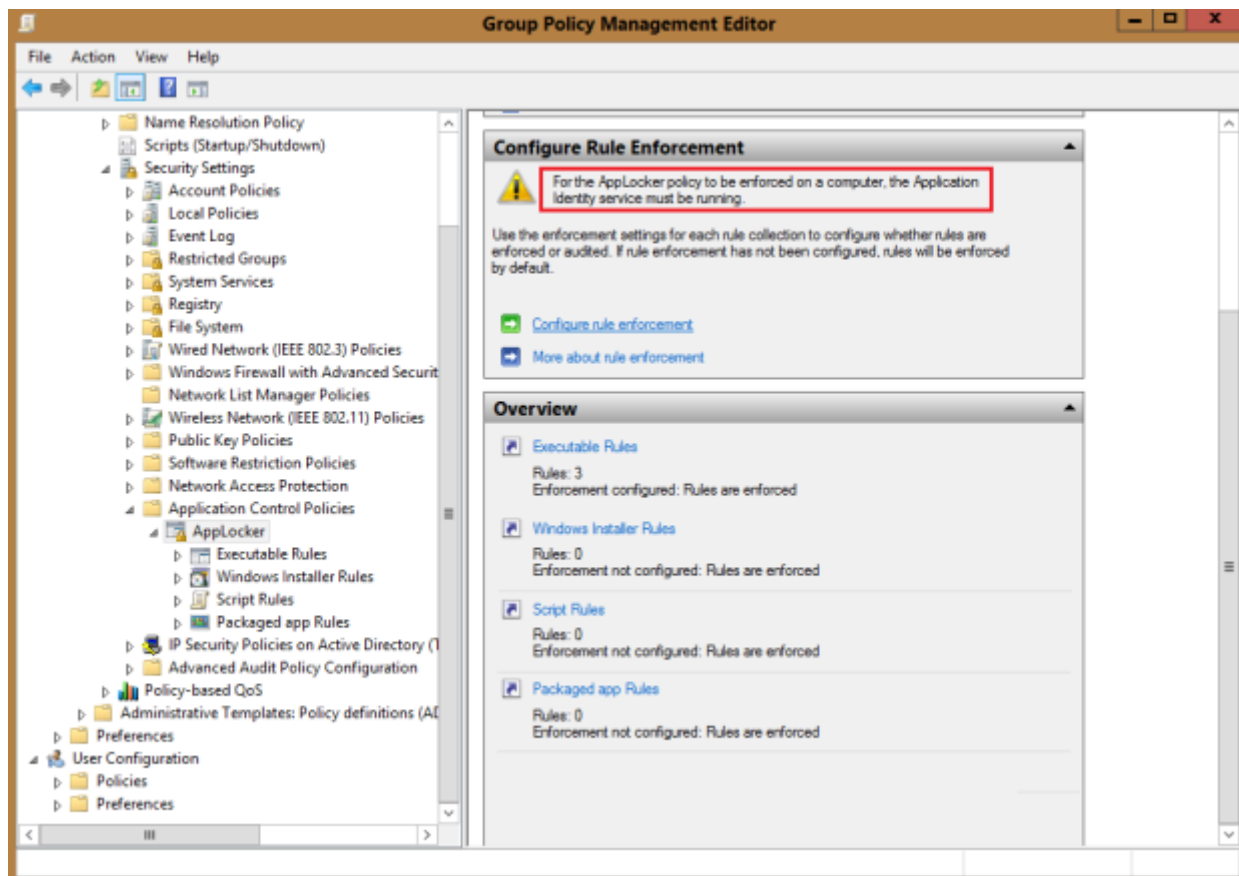
Enforce rules

[More about rule enforcement](#)

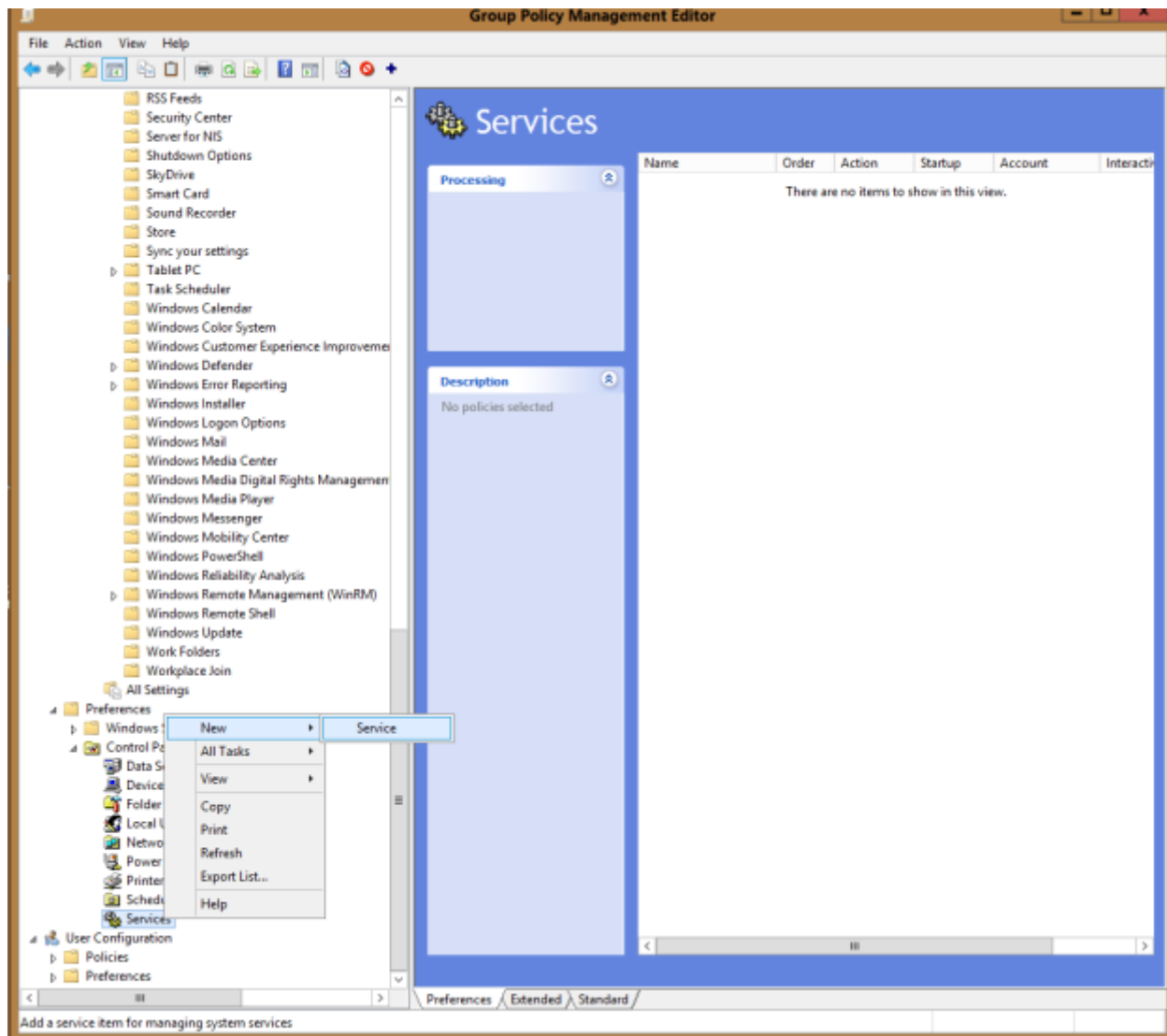
OK

Cancel

Apply




As for AppLocker policy to be enforced on a computer the *Application Identity* service must be running, let's add to the AppLocker GPO the enablement of the *Application Identity* service in the ...\\Preferences\\Control Panel\\Service section:



AppIDSvc Properties [X]

General Recovery Common

 Startup: No change ▼

Service name: AppIDSvc ...

Service action: Start service ▼

Wait timeout if service is locked: 30 ▲▼ seconds

Log on as:

☒ No change

☐ Local System account

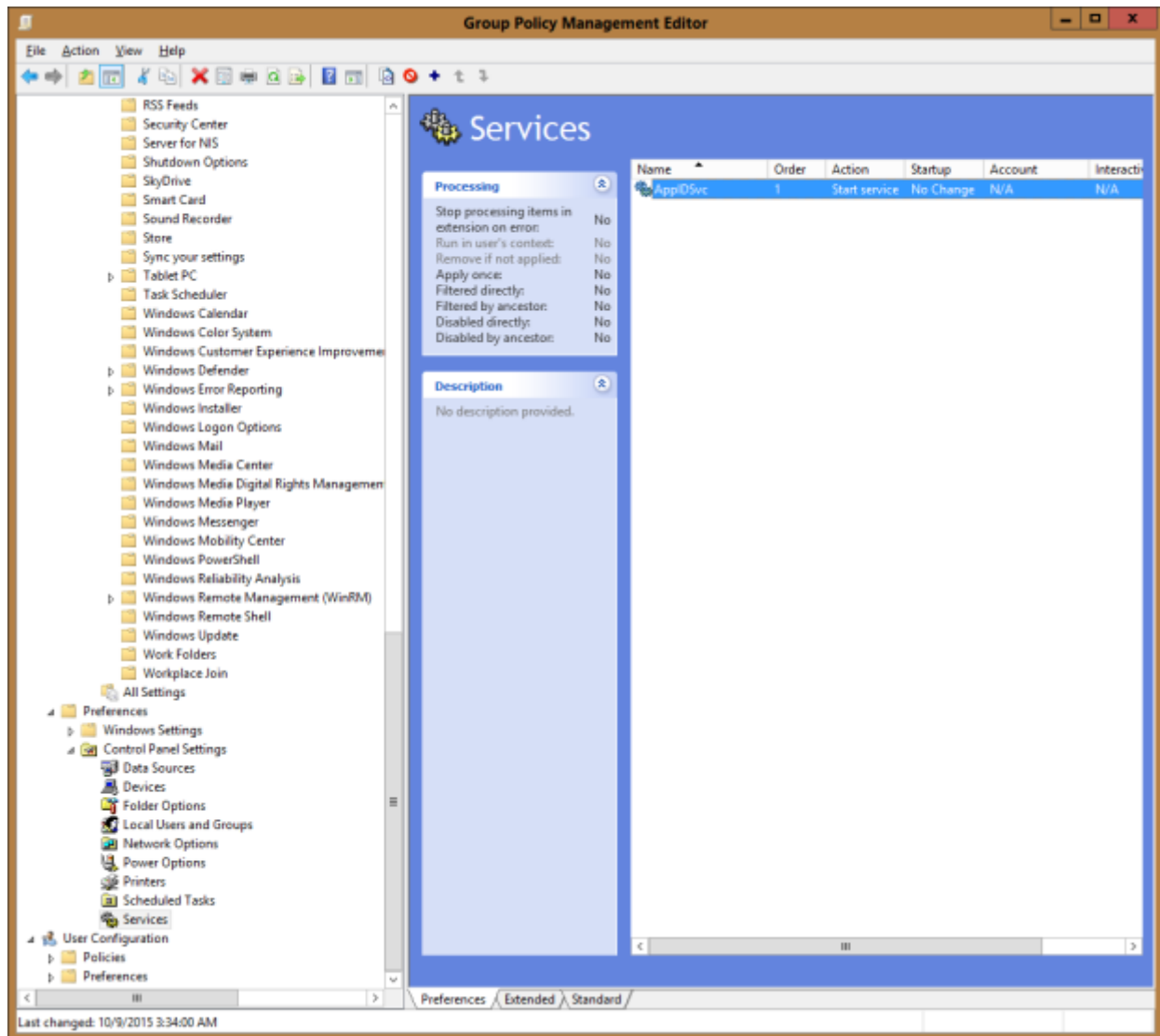
☐ Allow service to interact with desktop

☐ This account: ...

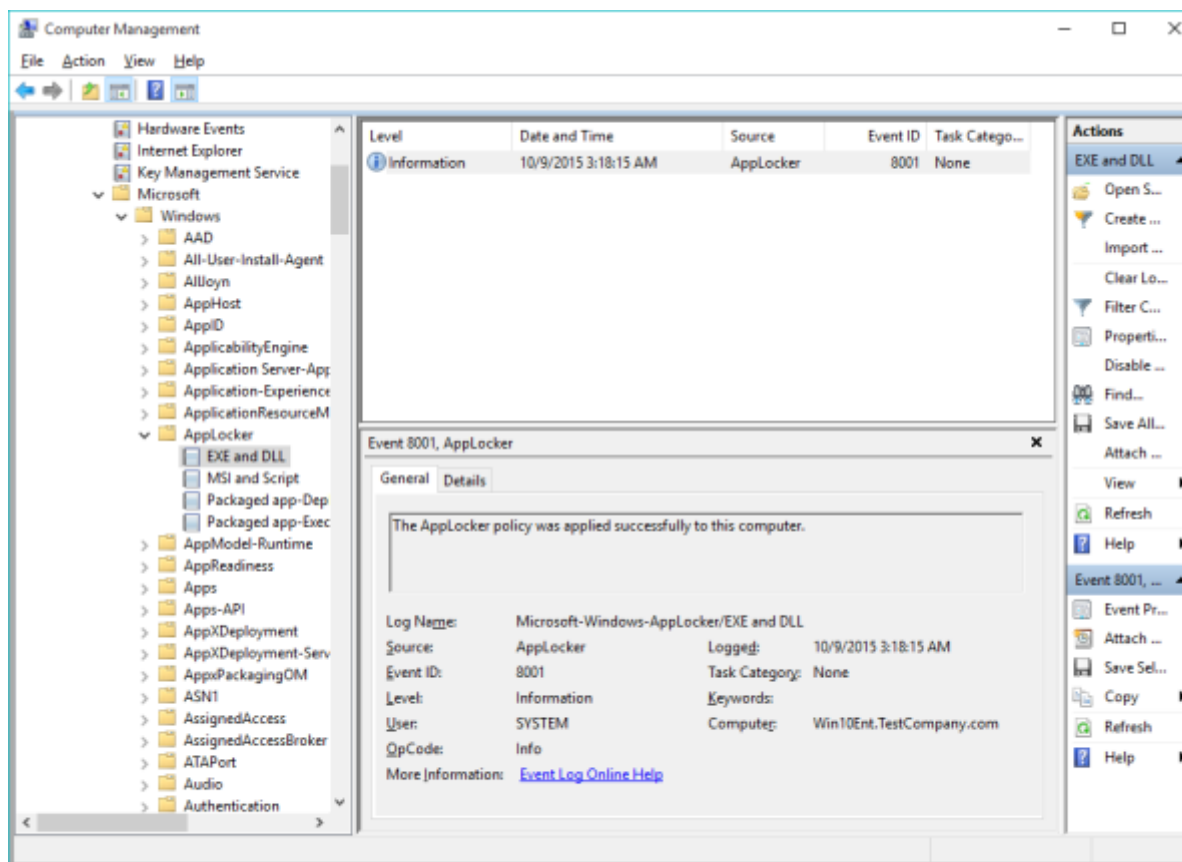
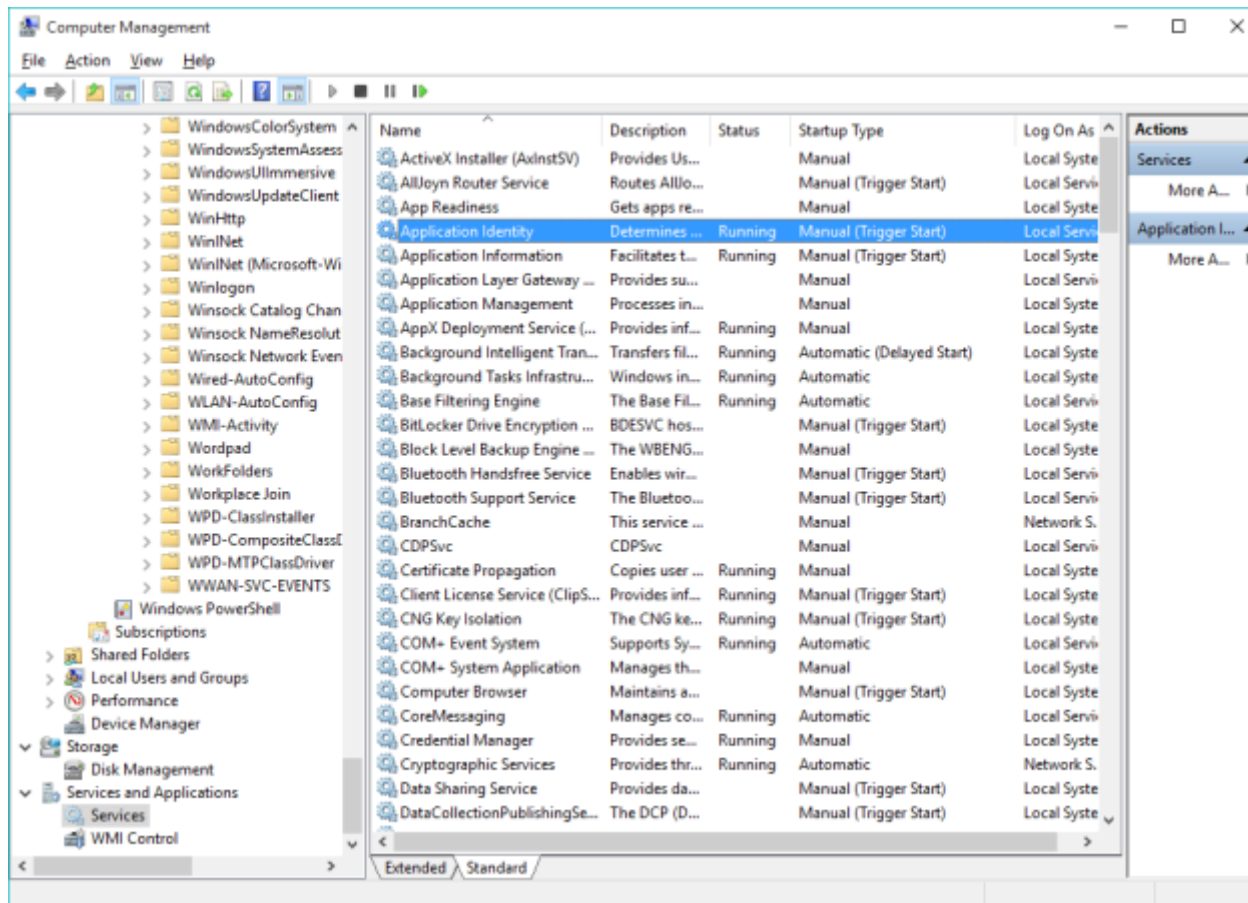
Password:

Confirm password:

OK Cancel Apply Help



After restarting my client Win10Ent (or running **gpupdate /force**) – up to two times as group policy might just be read after the first restart/gpupdate and only after the second be applied – the policy must be applied and *Application Identity* service must be running:



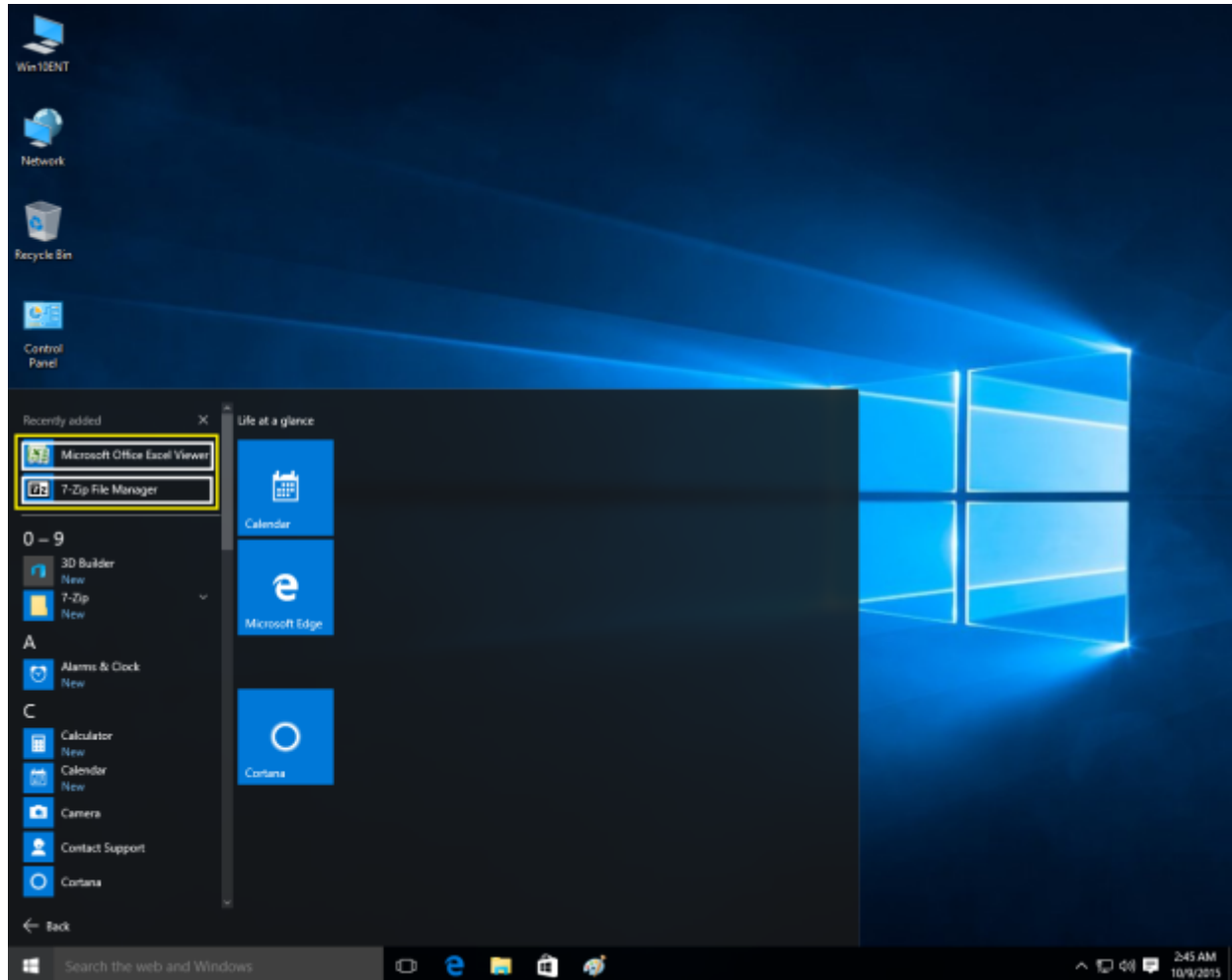
Now it's time to test the policy: I will try to do the following under I) **Domain\Admin** (TestCompany\ExAdmin) account II) **Domain\User1** (TestCompany\User1) account:

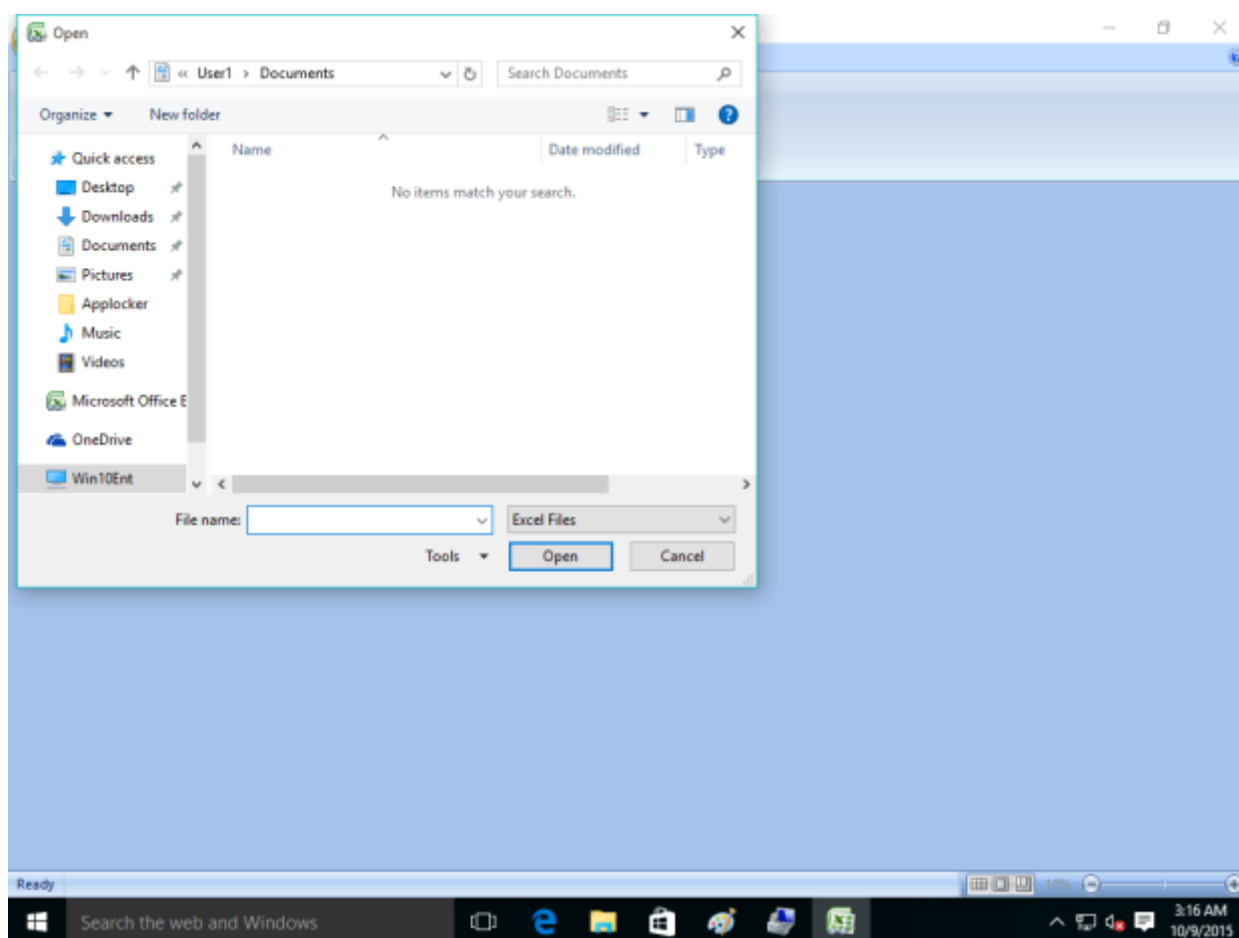
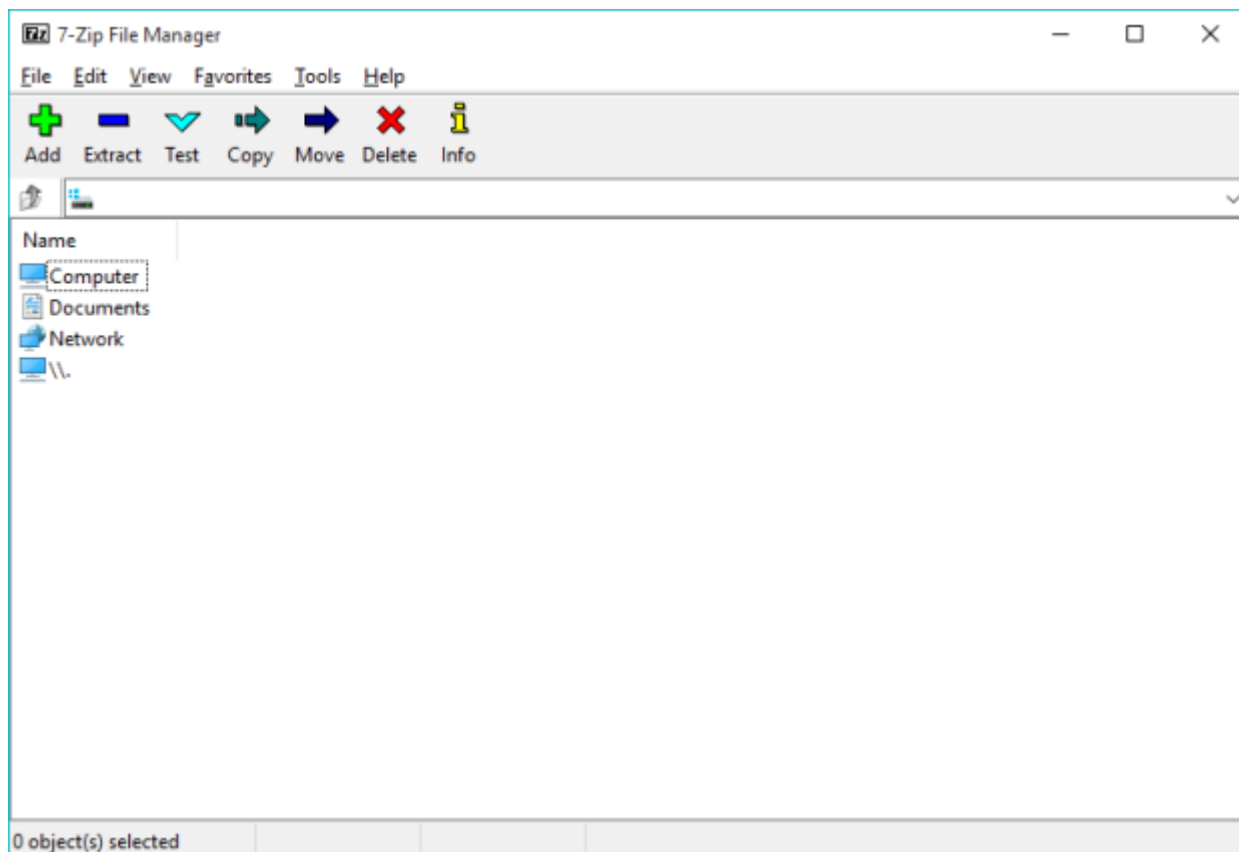
a) run allowed 7Zip

b) run MS Excel which was installed by the administrator and

* c) run built-in Calculator

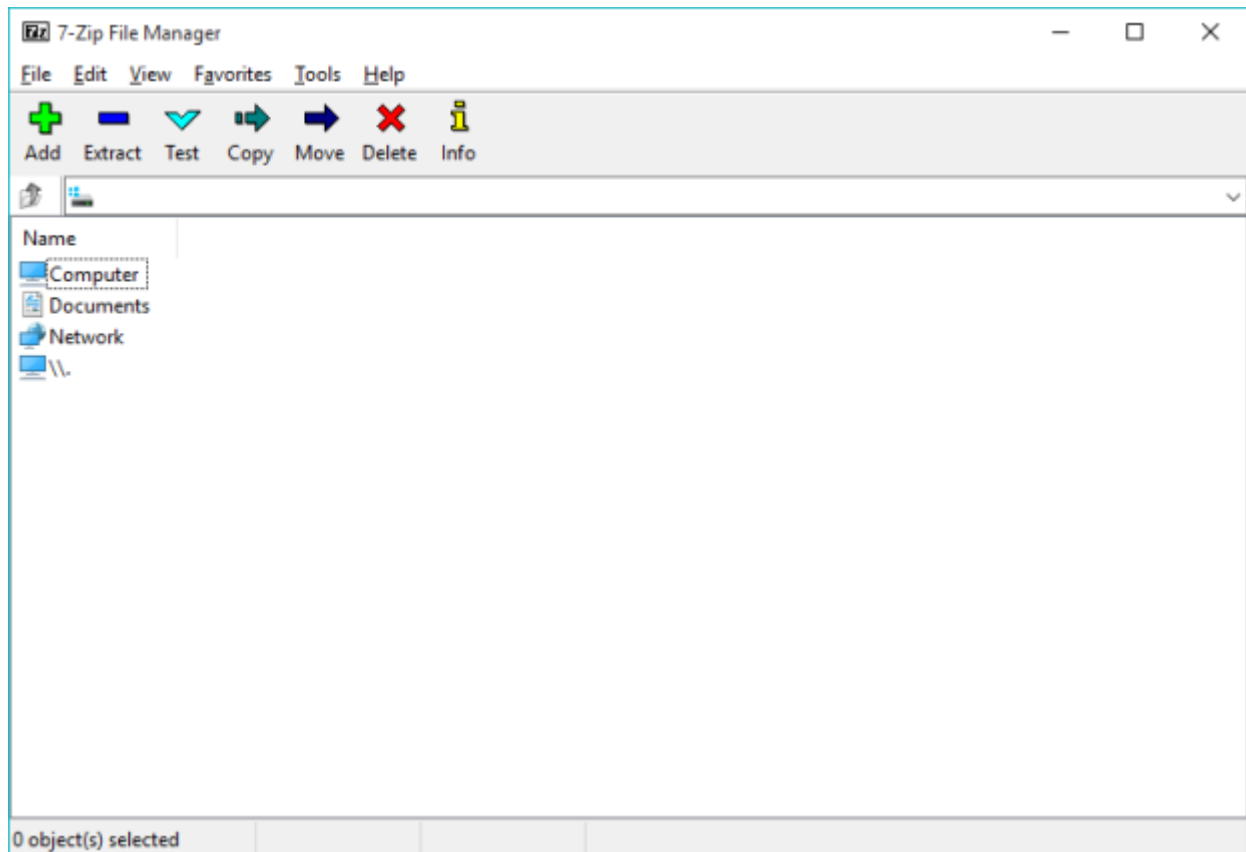
I-a) and I-b)



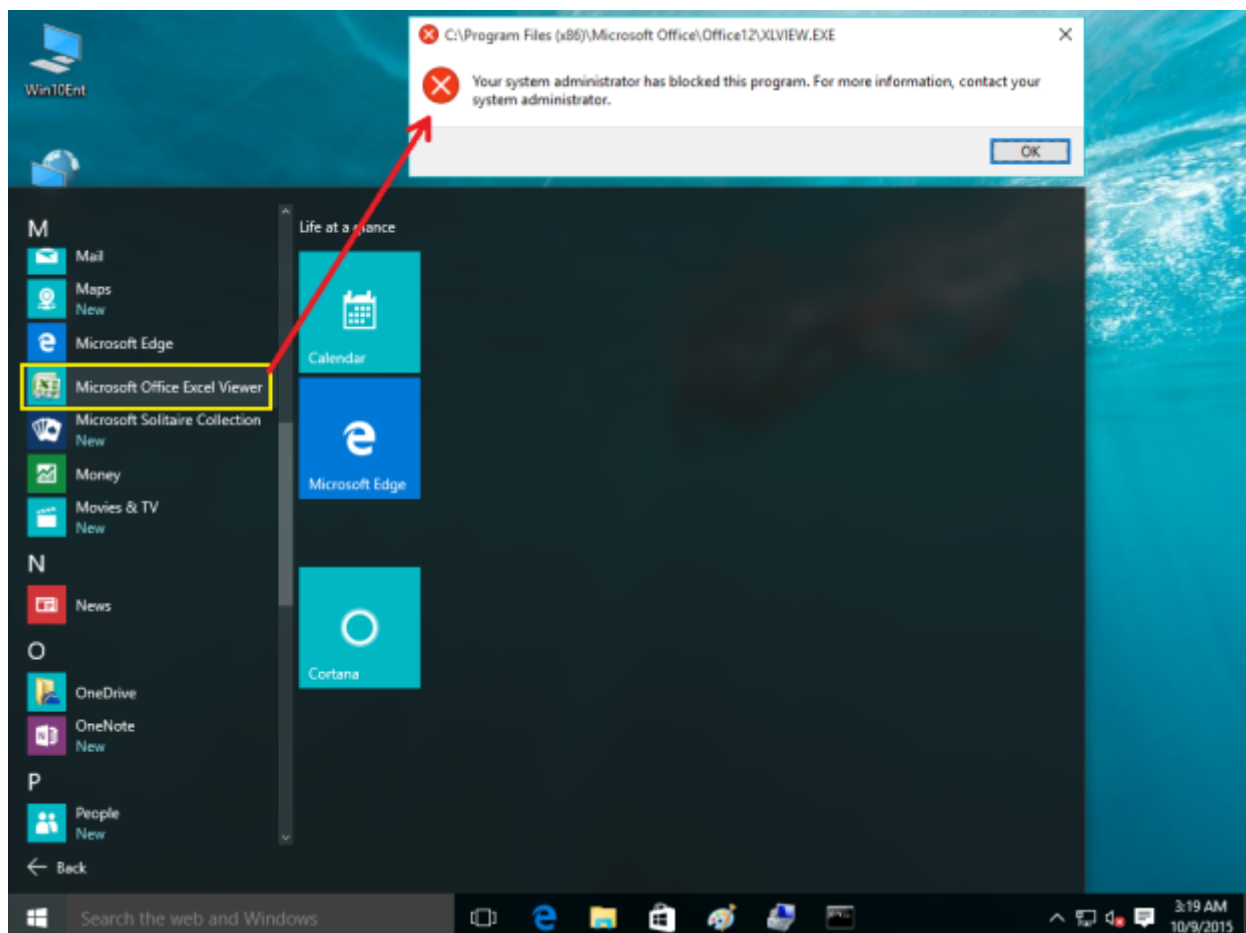


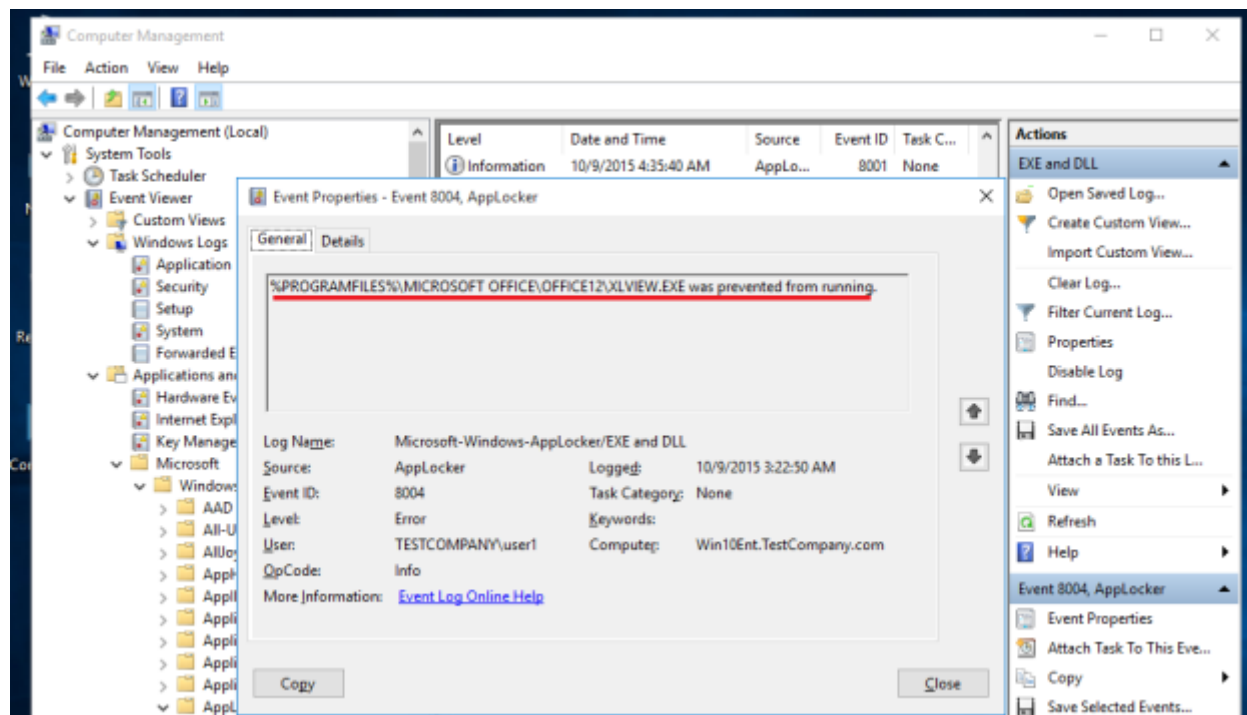
Both **7Z** and **MS Excel Viewer** open successfully because there's the AppLocker rule stating “**(Default rule) Built-in\Administrators – All folders.**” – exactly what I expected to see.

II-a) User1 can run 7Zip:



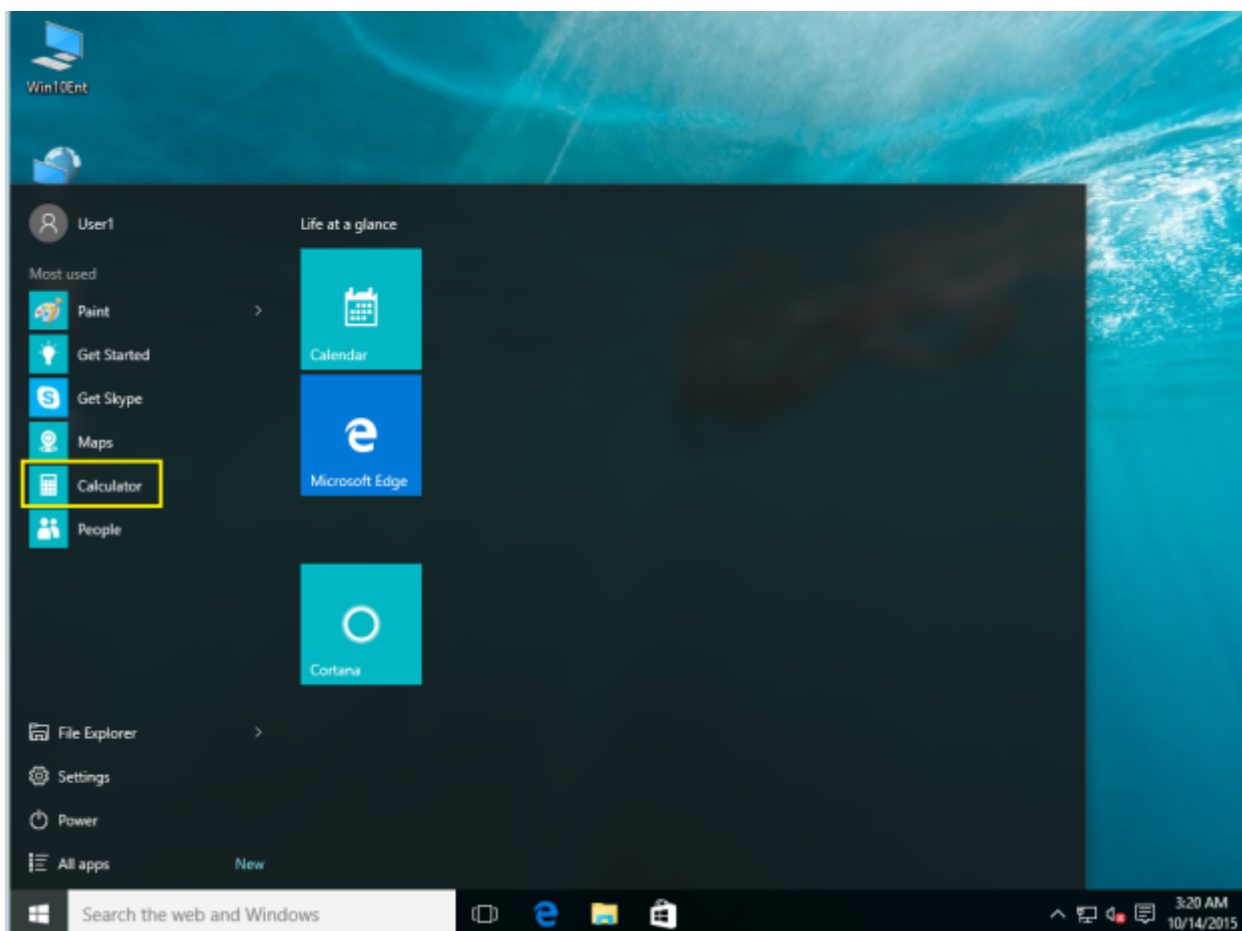
II-b) ...but can not run MS Excel Viewer:



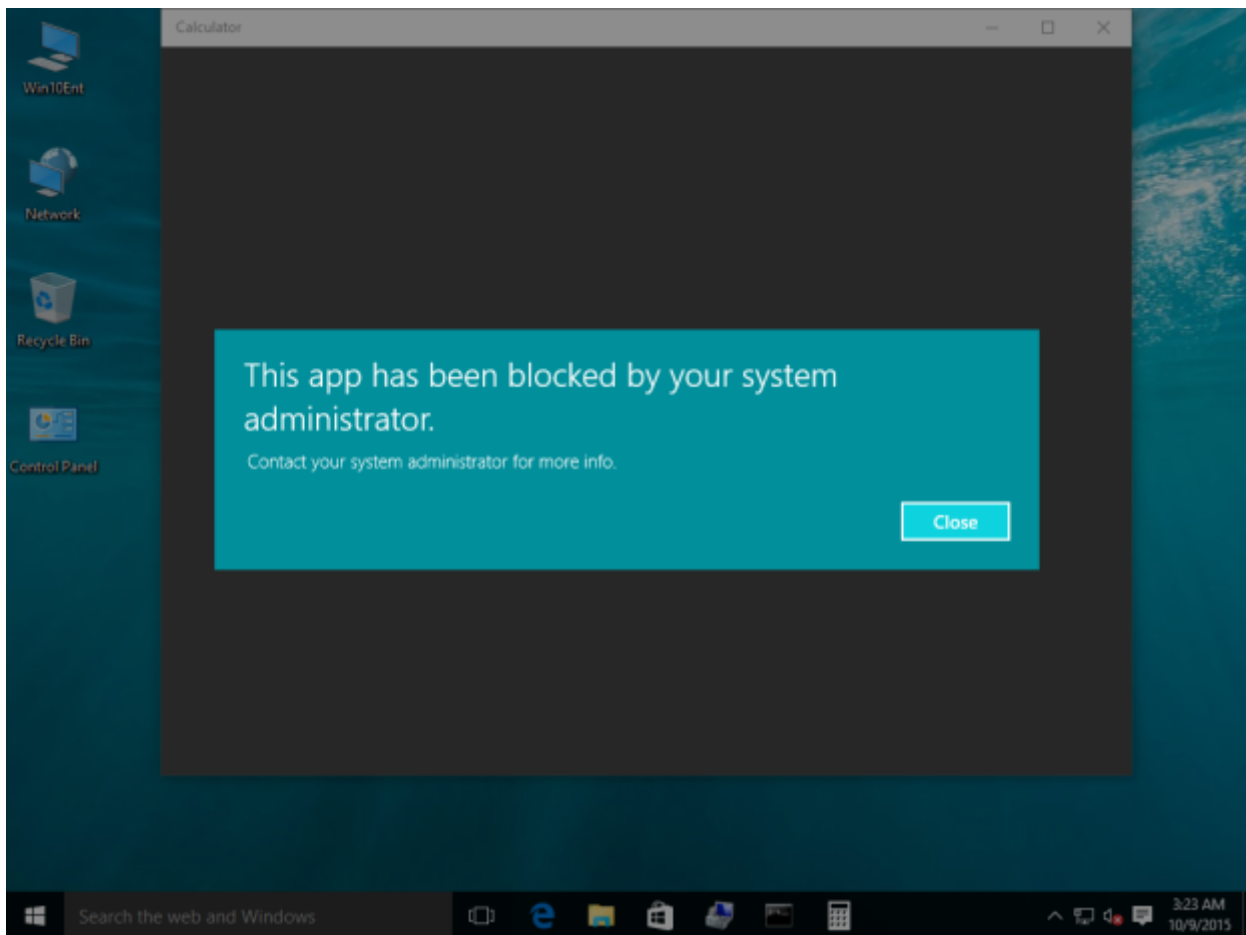
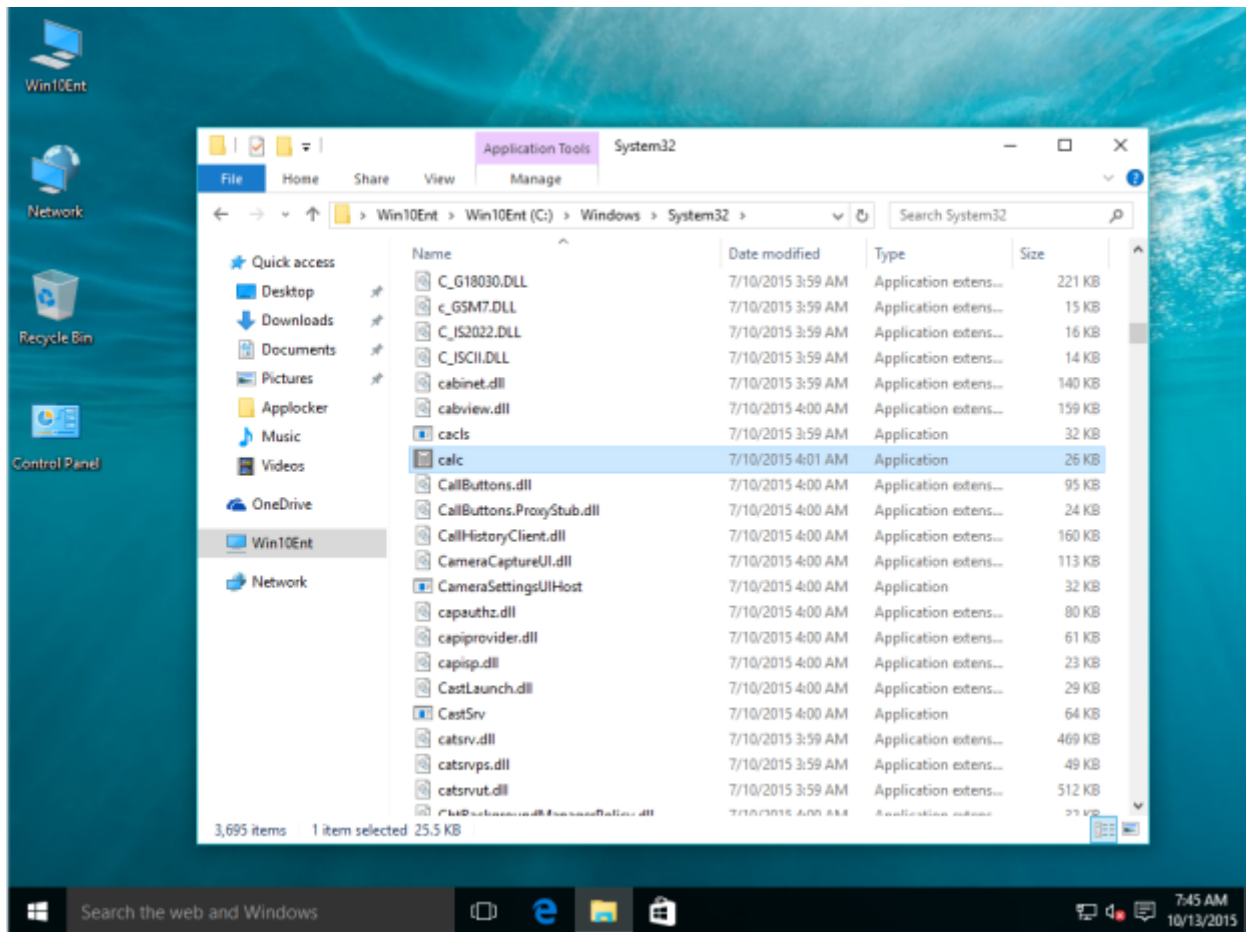


Again, all works as expected. But let's try to run the built-in Calculator as Admin and User1 and look at the results:

II-c) User starts the built-in Calc:



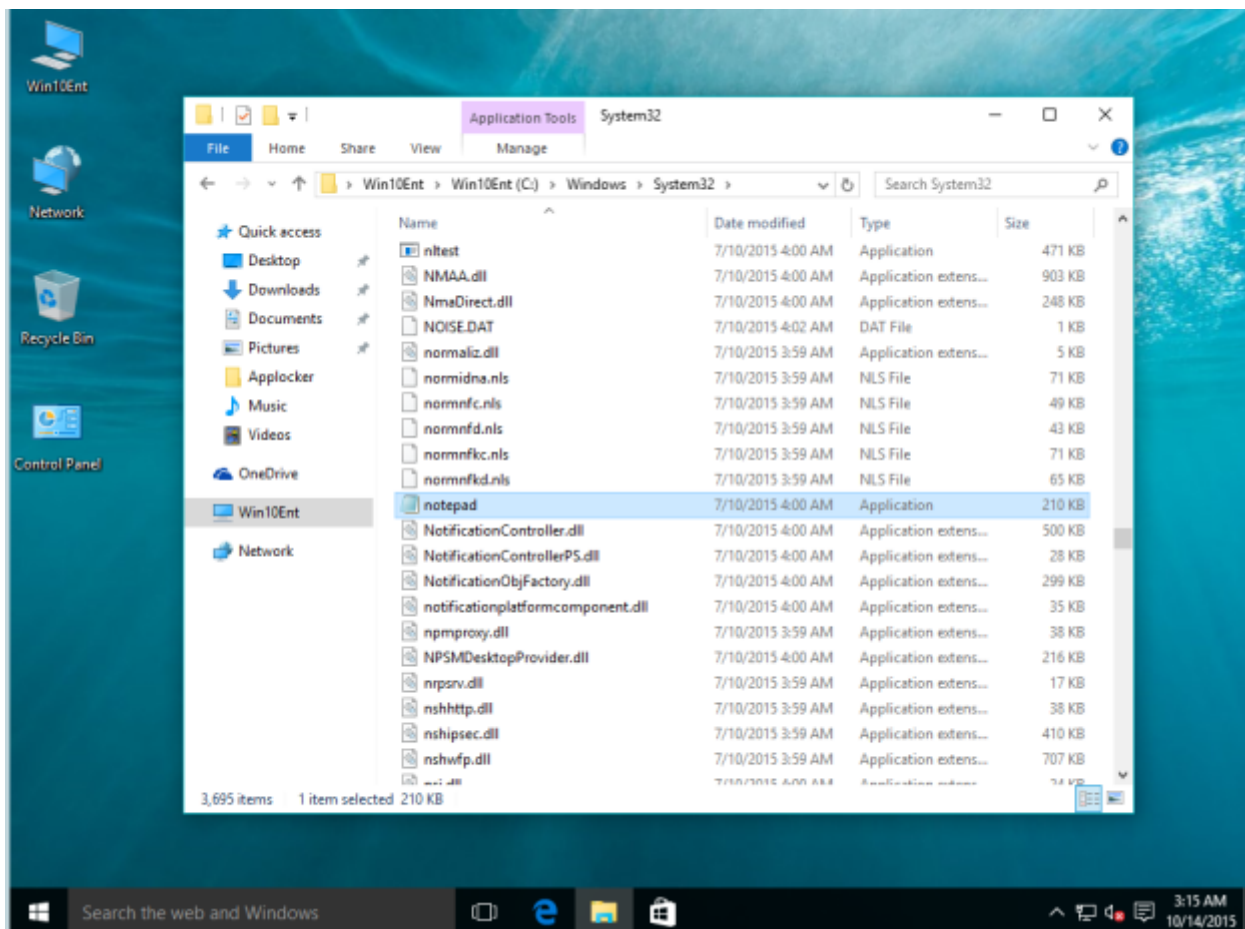
or by double-clicking the binary itself:

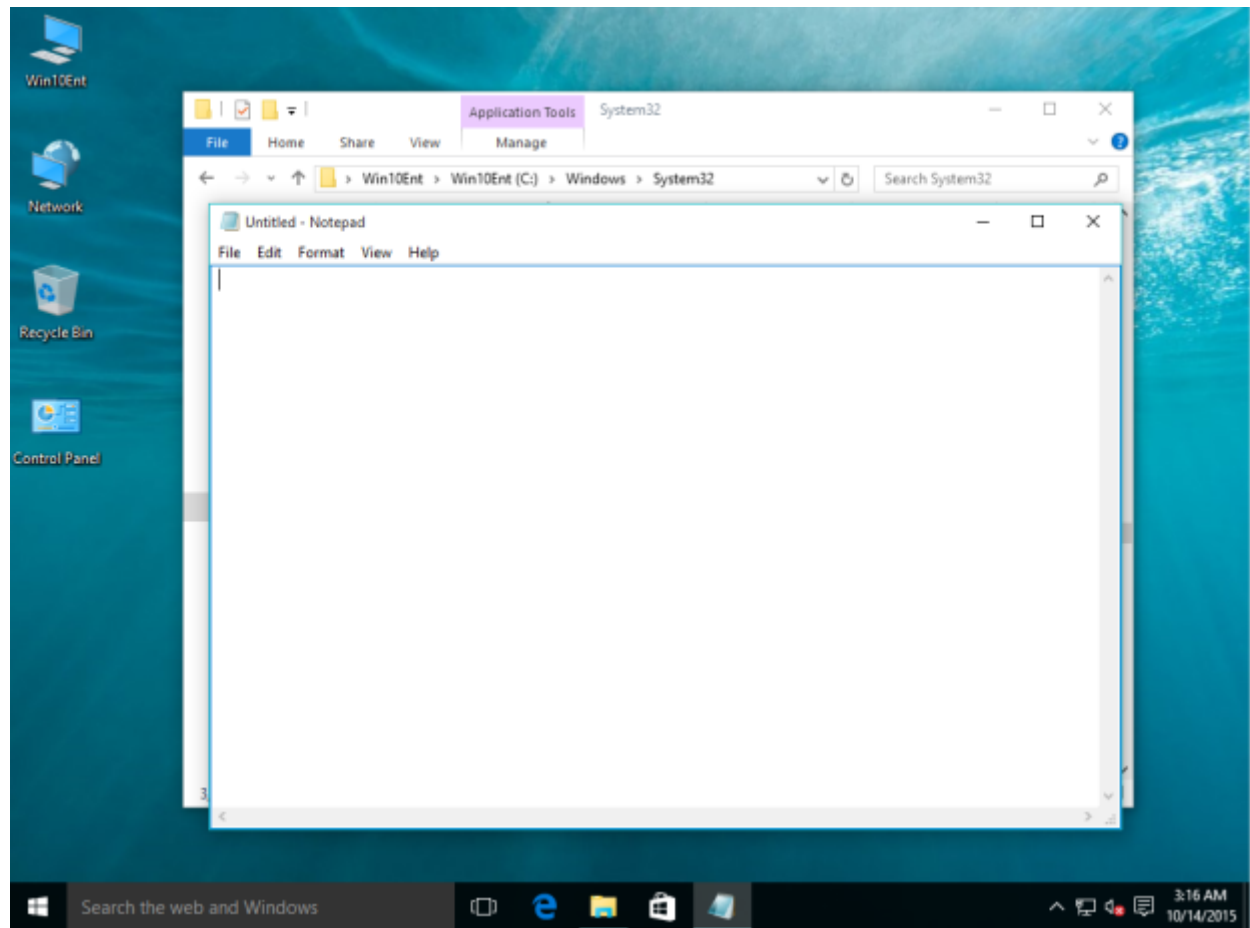


– there's the default rule allowing everyone to run any files located in the *Windows* folder so why User1 has not succeeded in running Calc.exe? If you take a closer look at the Calc item in the *Windows\System32* folder you will definitely notice that **Calculator** in **Windows 10** is a Metro-style app which will eventually be run from the *Program Files\WinApps* folder (for example you can run it by typing `CALCULATOR://` into the *Run* box), the Calc.exe in the System32 folder is just a wrapper that refers to the *Program Files\WinApps\Microsoft.WindowsCalculator_...* subfolder.

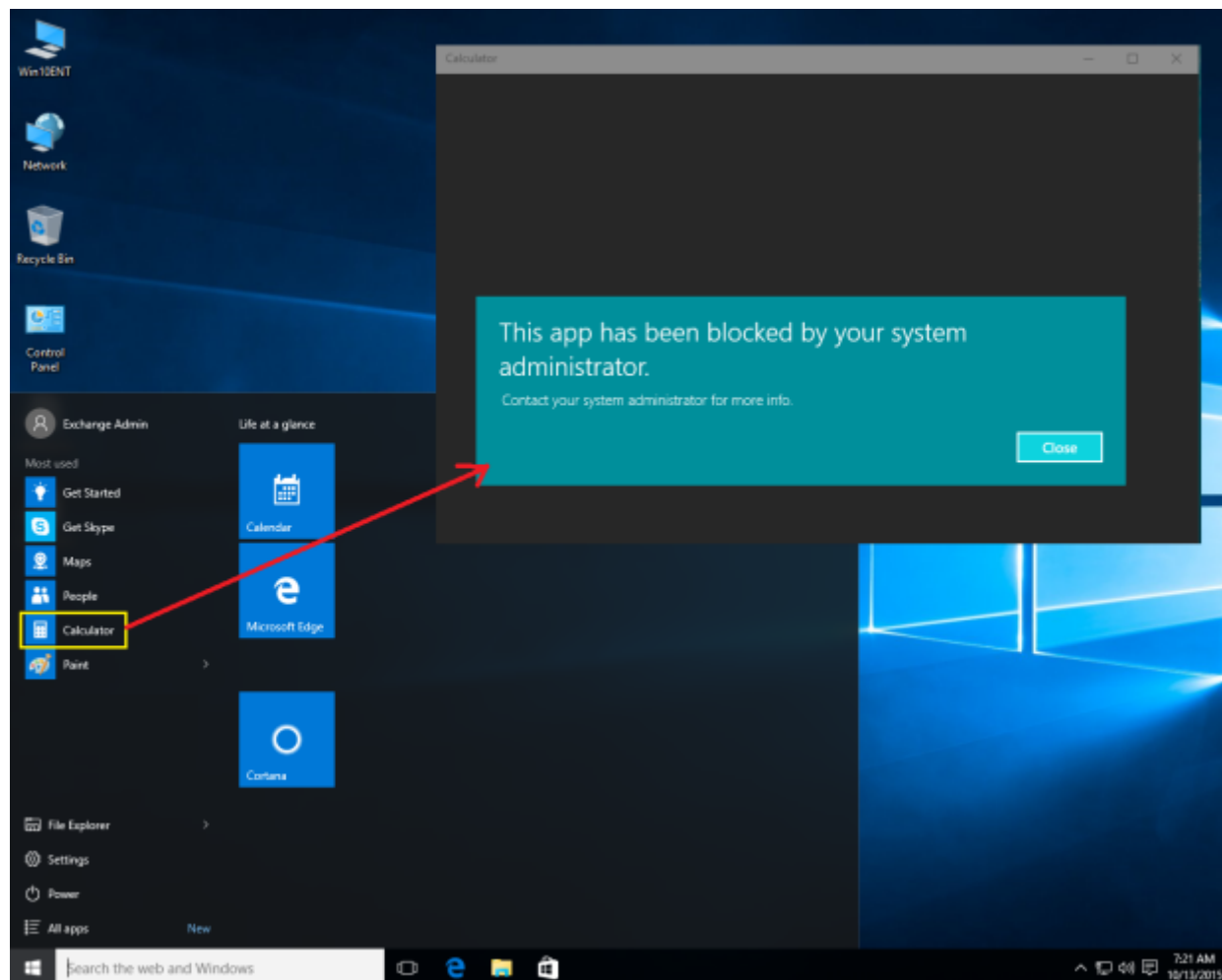
Given that there's no AppLocker rule that would allow *TestCompany\Domain Users* to run programs from WinApps folder this behaviour is by design.

We can make sure this is the case by running Notepad which is the old classic application from the same *Windows\System32* folder:

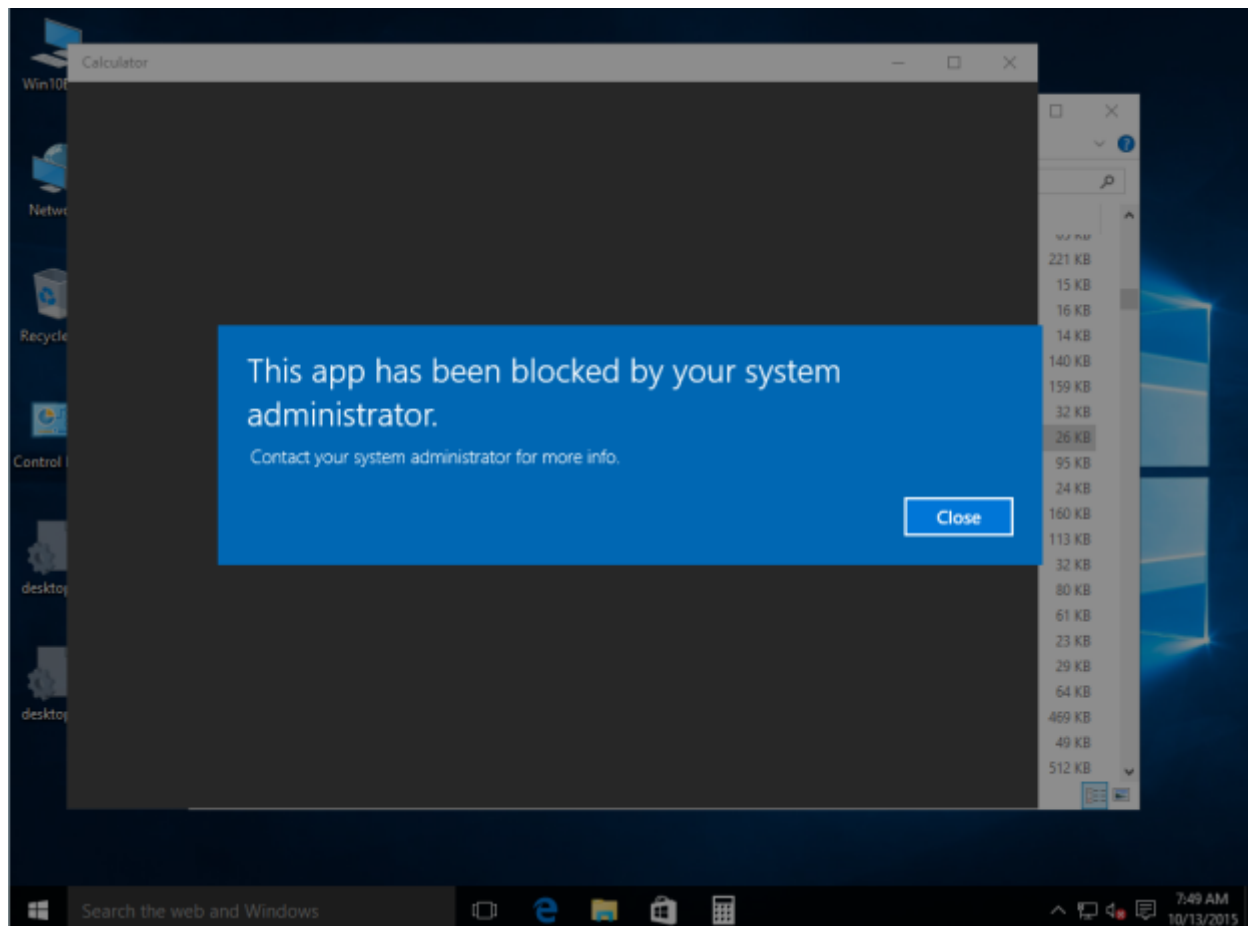
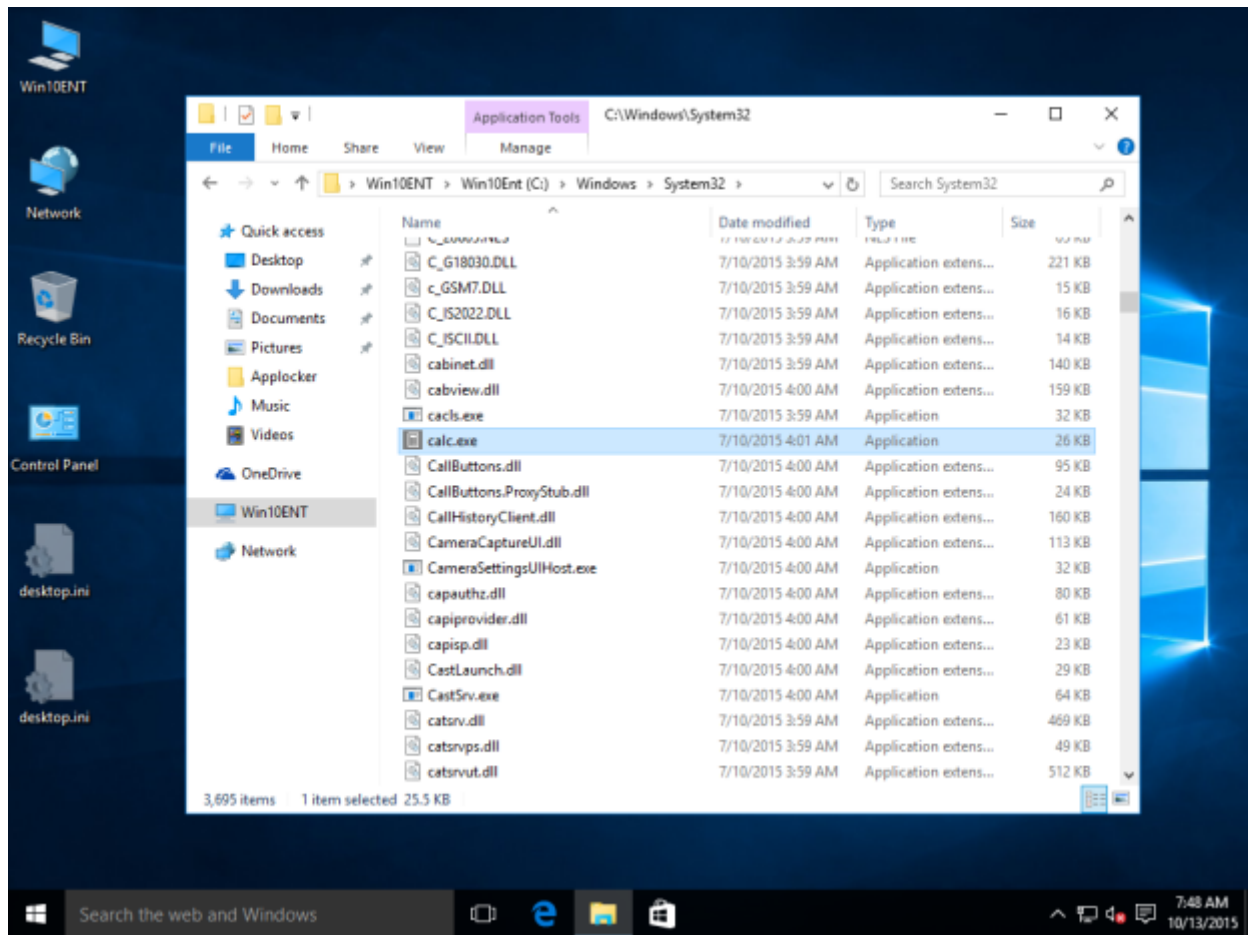




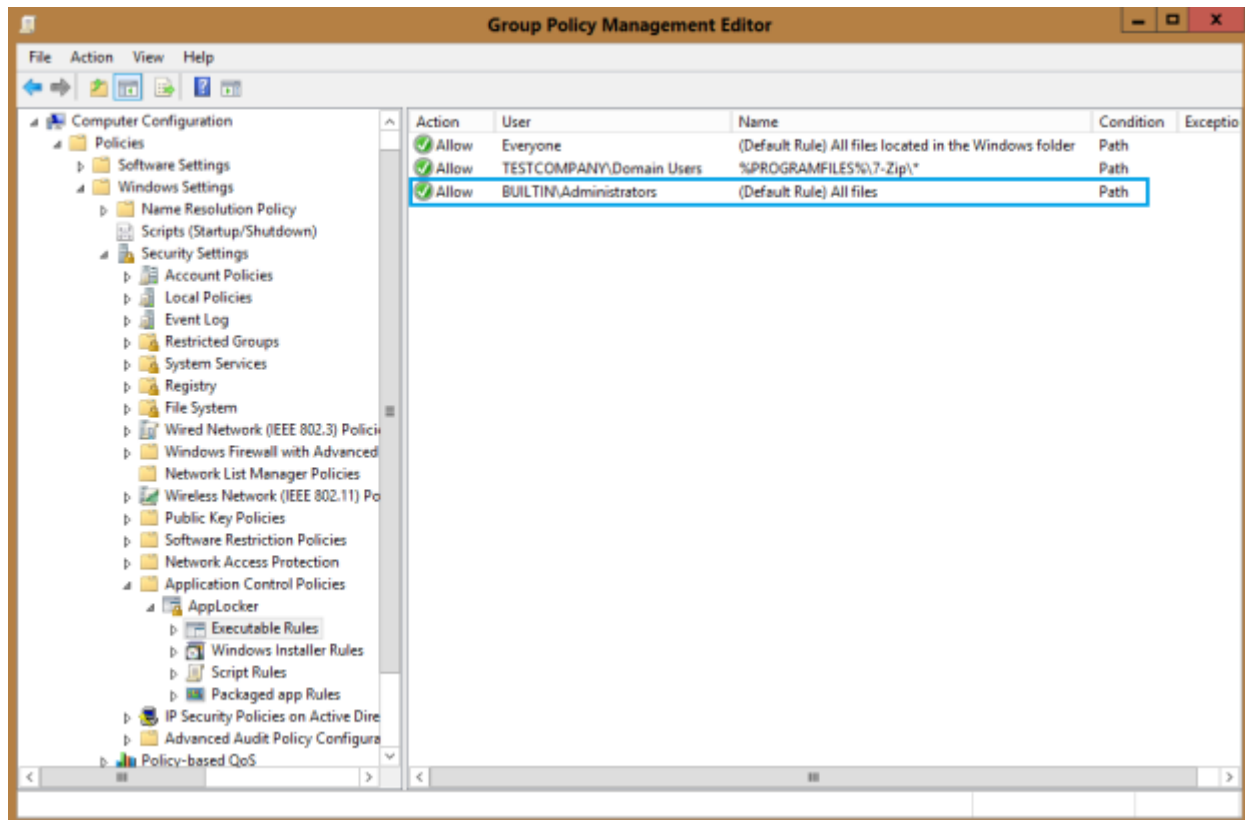
I-c) Administrator starts the built-in Calculator:



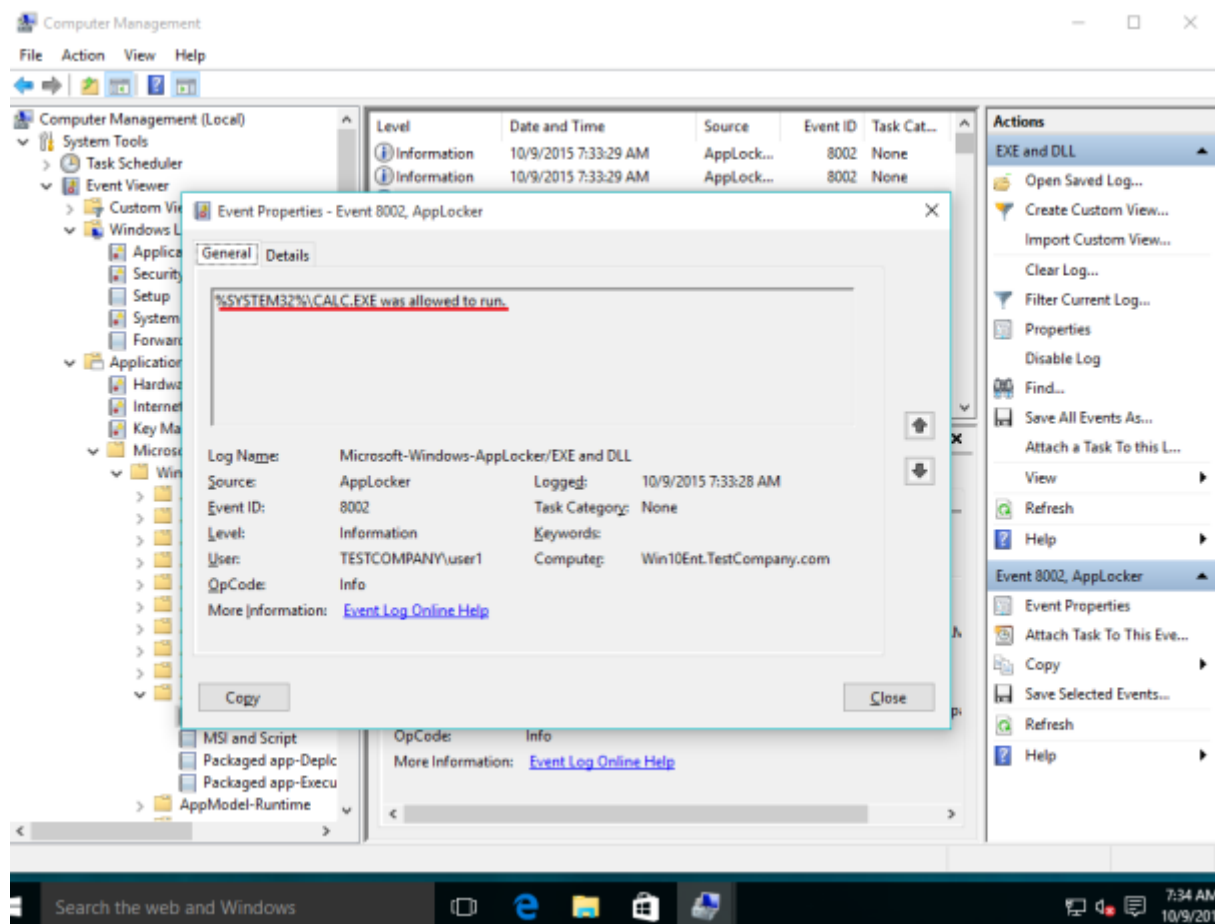
..or from *Windows\System32*:



Although there's the explicit allow rule for *Built-in\Administrators* for all folders (including *Program Files\WinApps*) this administrator (*TestCompany\ExAdmin*) could not run the built-in **Calculator**.



Furthermore, if we look in the AppLocker log we'll see that blocking Calculator produces the "Allow" event (both for Administrator's and User1's accounts):



Whatever the reason is for blocking the built-in Calculator Windows should not register “allow” event 8002 instead of “deny” 8004 event – this is the first bug in **Applocker/Windows 10 Enterprise** configuration.

The second is the obviously incorrect Applocker handling of the metro style applications, which leads not only to blocking allowed applications but to non-functioning Start menu and desktop customizations as well (cause they’re the metro style apps themselves).