Cybersecurity

Subject: Windows network security

Windows network presents some distinctive characteristics, and due to its prevalent role in the networking market it is compulsory for us to be aware of their most significant security implications. The objective of the following challenges is to acquaint the student with those issues.

1. Windows network protocols

1.1 Network environment

SMB (*Server Message Block*) is the network protocol supporting the resources shared in the network environment in the MS Windows. Unfortunately, it is not a sophisticated protocol, especially in terms of security.

1.1.1 Shares

In the SMB protocol terminology, *shares* are the operating system resources shared over the network. The `net` command with the `share` argument displays the list of current local system shares:

```
C:\>net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
C$           C:\                             Default share
IPC$                                         Remote IPC
ADMIN$       C:\Windows                      Remote Admin
Public       C:\Public
The command completed successfully.
```

In the example above, the UNC path to the `C:\PUBLIC` resource shared on `virtual` would be `\\virtual\PUBLIC`. The share name does not need to match the resource name on the local file system.

The resources can be made available to the public (without the need for authentication—the Everyone group) or to specific local users, domain users, or e.g. the dynamic Authenticated Users group (including any properly authenticated local or domain user identical to the user of the account requesting access). The SMB protocol performs authentication as transparently as possible—first it tries to authenticate the current local user with his current password to the remote system. If the attempt is unsuccessful, the operating system prompts for a username and password.

By default, the Everyone group is granted read permissions to a newly created disk share.

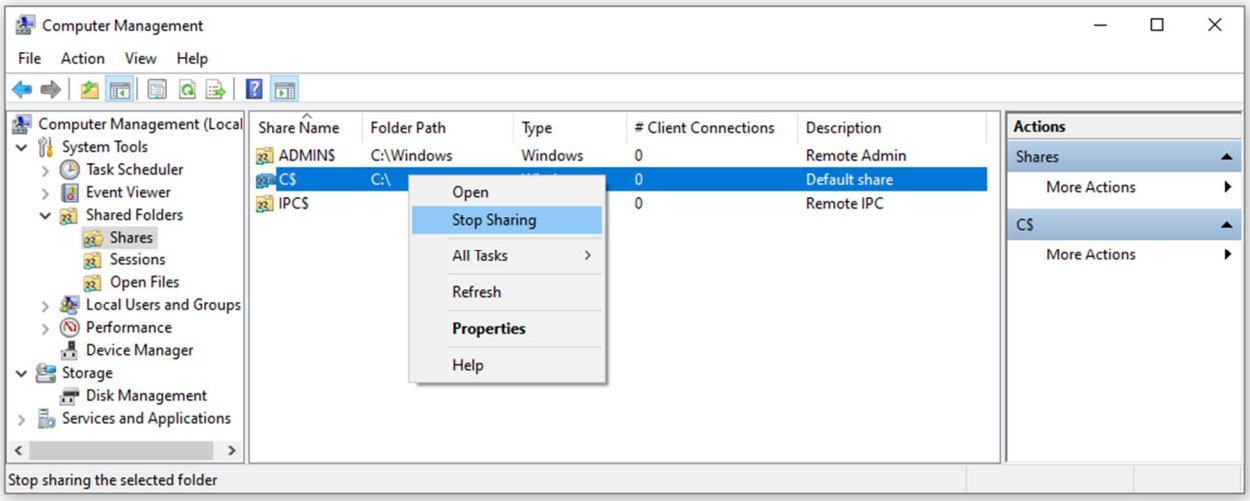
The list of available shares in the entire local Microsoft Network or in a specific system is displayed by the `net view` command, e.g.:

```
C:\> net view \\virtual
```

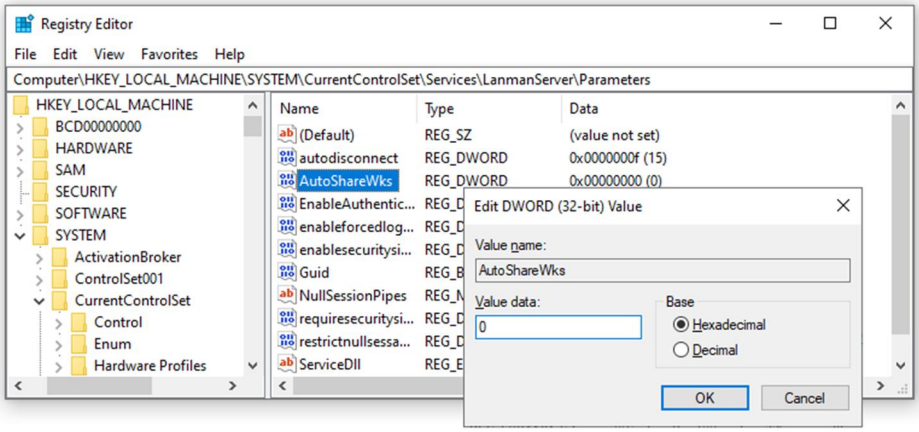
MS Windows systems, immediately after installation, create some default shares (e.g. C$ and ADMIN$), the existence of which usually, especially in the case of individual users, is not needed and constitutes an additional potential risk. You can disable these shares using the `net share` command, e.g .:

```
C:\> net share C$ /delete
```

or using the Computer Management console:



To permanently remove these shares, you must modify the system registry. In the HKLM\SYSTEM\CurrentControlSet\Services\LanManServer branch, in the Parameters key, add a value named AutoShareWks of DWORD type equal to 0:



After such registry modification and system reboot, default shares are disabled.

1.2 Hiding computer in network environment

In Windows, it is possible to disable the visibility of the name of a given system in the network environment, while still preserving the access to its shares. To do this, the `net` command can be used:

```
C:\> net config server /hidden:yes
```

For this operation to be effective, Windows documentation suggests disabling also the following system services: *SSDP Discovery Service*, *Computer browser* and *UPnP Device Host*. Practice shows, however, that instead of them, the *Function Discovery Resource Publication* service should be deactivated.

1.3 Network connections

Detailed information about active network communication (e.g. active services, established connections, network traffic statistics) can be obtained by using the `netstat` command:

```
C:\> netstat -an
```

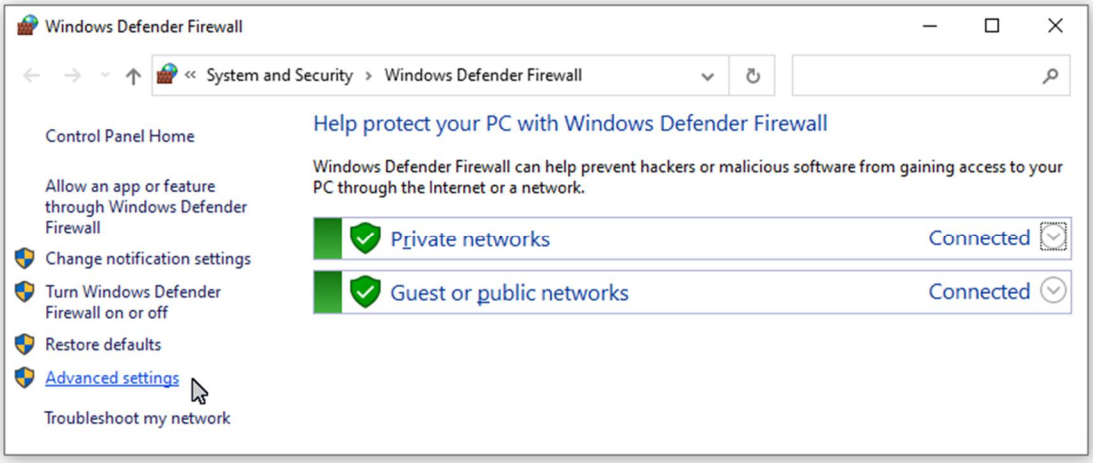
`a` – all active connections + TCP and UDP ports (on which computer listens)
`n` – displays addresses and port numbers numerically (no names are determined).

```
C:\>netstat -na

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    10.0.2.15:139          0.0.0.0:0              LISTENING
  TCP    10.0.2.15:49736        213.216.115.49:80      ESTABLISHED
  TCP    10.0.2.15:49739        2.20.28.114:443        ESTABLISHED
```

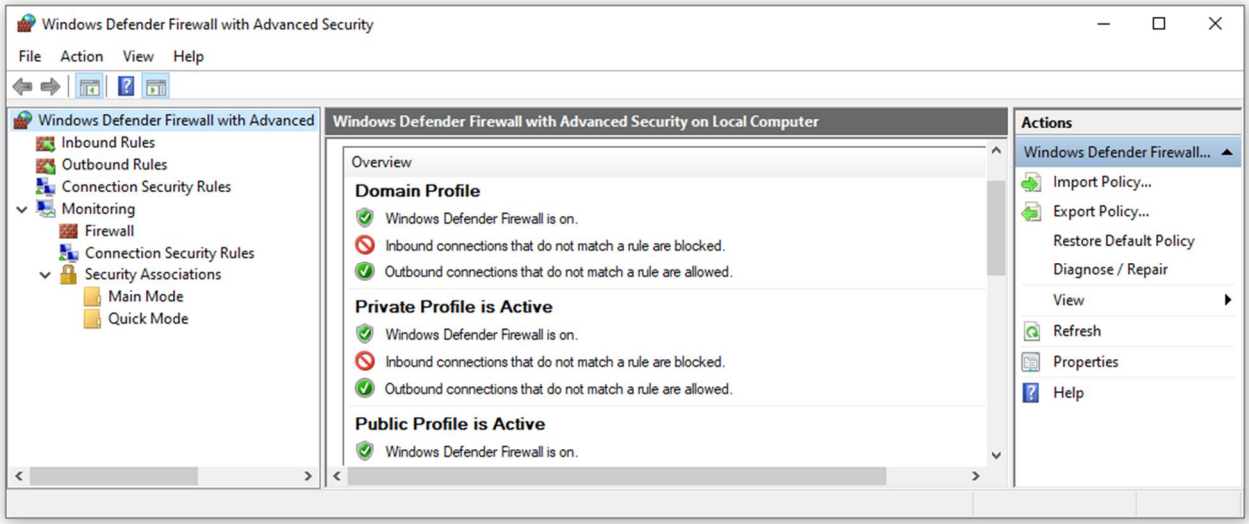
An useful parameter is b, revealing processes holding a connection or listening to a port.

2. Network firewalls

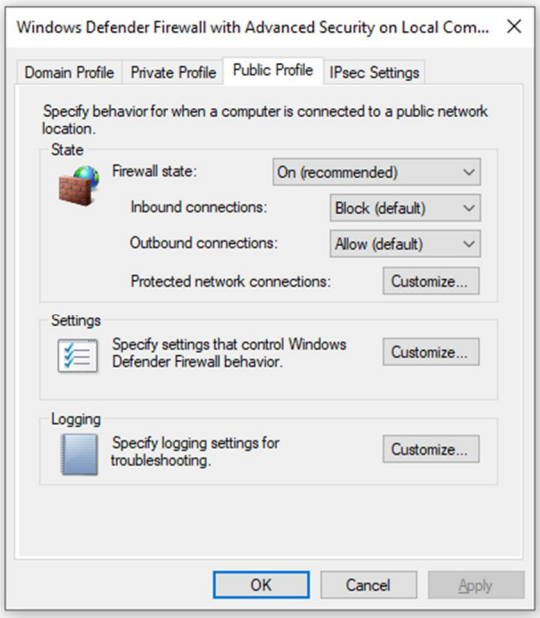
Windows has a built-in personal firewall. The firewall can be set-up, using the Control Panel, Network and Sharing Center, or via the Local Security Policy console. Advanced settings section are particularly important for us:

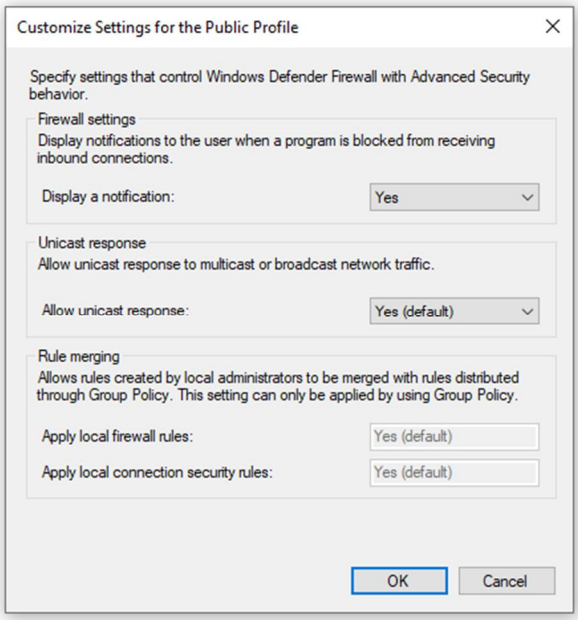


Settings are divided into 3 network communication profiles: *domain*, *private* (both for private networks) and *public* (for public networks):

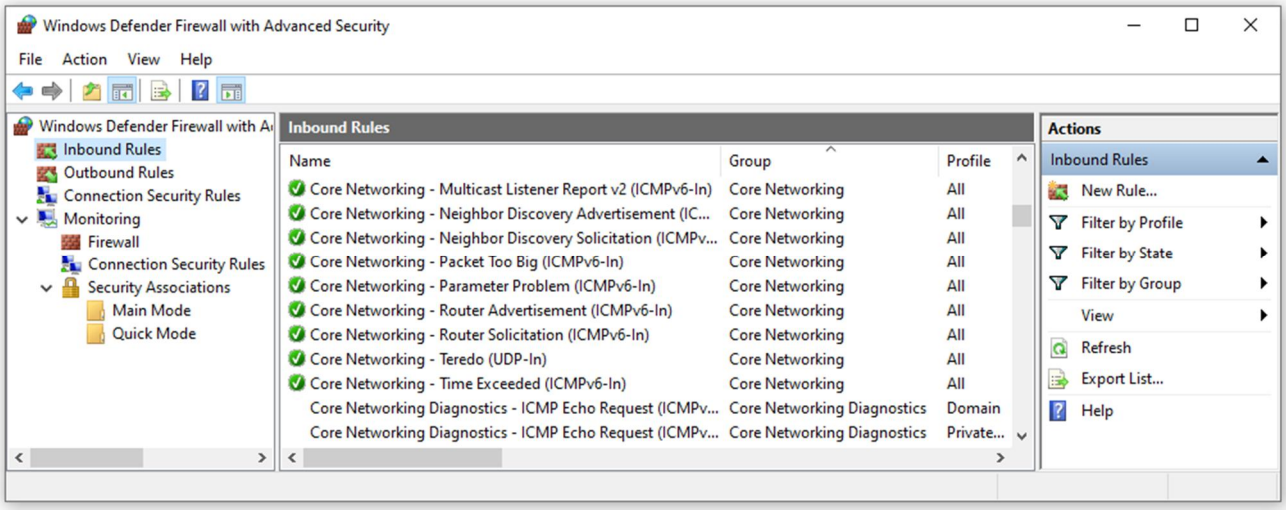


For each profile you can define, among others, the default-action policy, or alert and logging rules:





By default, the firewall provides some ready-to-use filtering rules defined separately for incoming and outgoing traffic:



Each rule defines the filtering criteria and one of three possible actions:

- block,
- allow,
- allow the connection if it is secured by IPsec (which will be a separate topic for us).

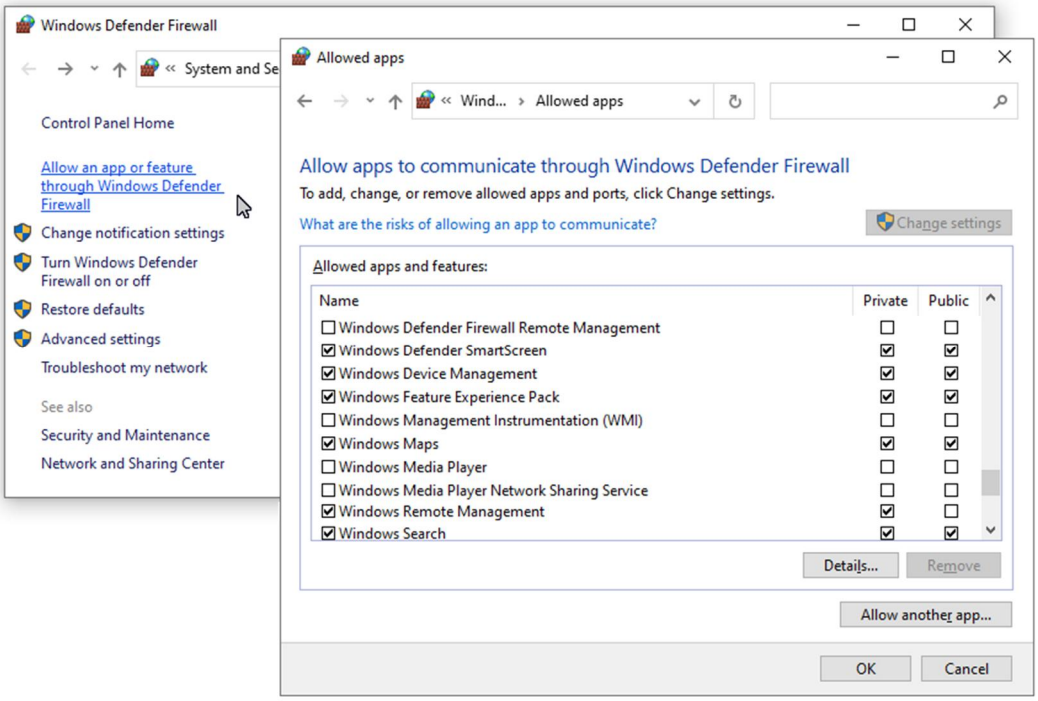
Alongside comfortable graphical interface, firewall rules can also be managed using netsh shell. For instance, a simple new filtering rule can be created with the command:

```
C:\> netsh advfirewall firewall add rule name="All ICMPv4"
protocol=icmpv4 dir=in profile=private action=allow
```

or:

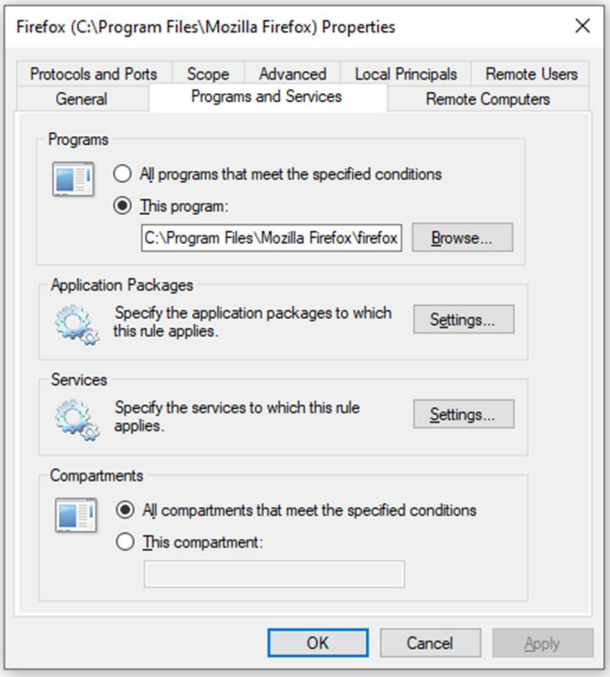
```
C:\> netsh advfirewall firewall add rule name="ICMPv4 echo"
protocol=icmpv4:8,any dir=in profile=public action=block
```

It is also possible to define global permissions per program:



which can be further used in the firewall policy settings for individual profiles:

A particular program can also be specified in a filtering rule:



or with the netsh shell:

```
C:\> netsh advfirewall firewall add rule name="my rule" dir=in
action=allow program="C:\...\httpd.exe" protocol=TCP localport=80
remoteip=192.168.56.1,172.16.0.0/16,LocalSubnet profile=domain
enable=yes
```

2.1.1 Monitoring firewall operation

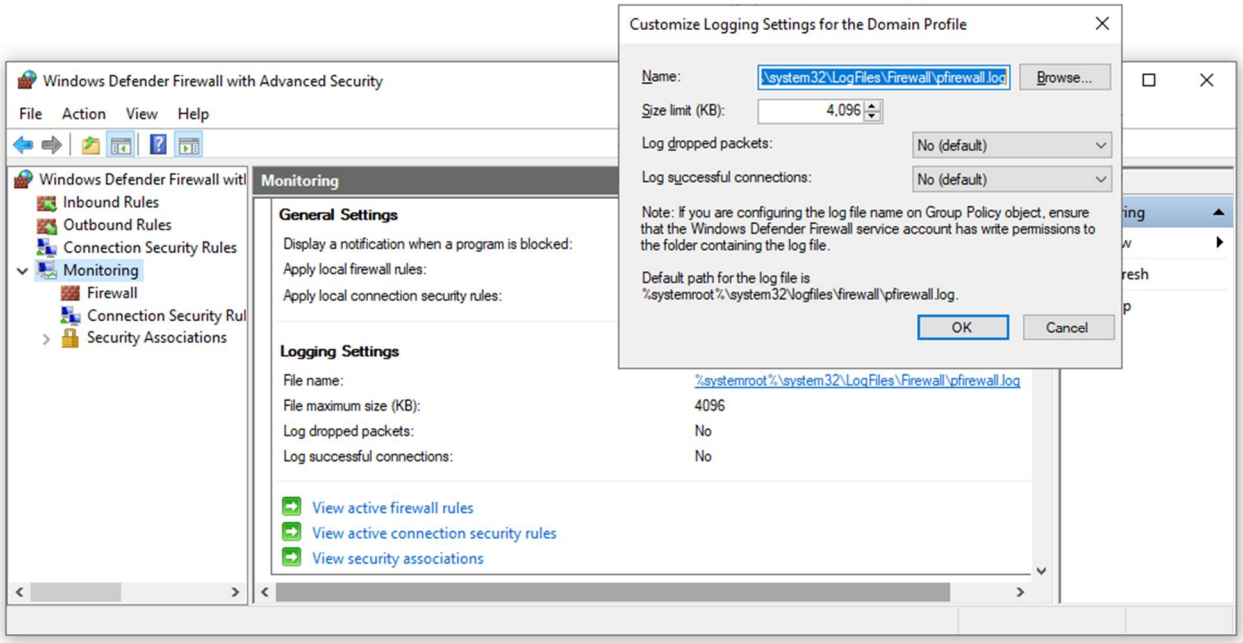
The netsh shell can also be used to set the event monitoring parameters for the firewalled traffic:

```
C:\> netsh advfirewall set currentprofile logging filename
%systemroot%\system32\LogFiles\Firewall\pfirewall.log

C:\> netsh advfirewall set currentprofile logging maxfilesize 4096

C:\> netsh advfirewall set currentprofile logging droppedconnections
enable

C:\> netsh advfirewall set currentprofile logging allowedconnections
enable
```

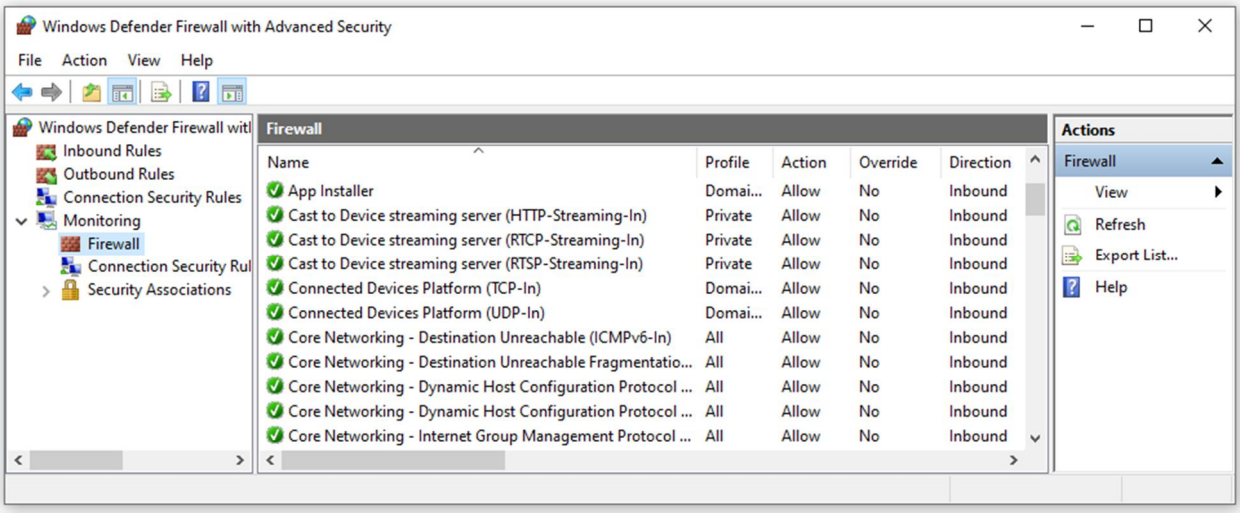
The log has a text file format with the structure shown in the exemplary content:

```
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn
tcpack tcpwin icmptype icmpcode info path

2013-04-24 10:34:30 DROP ICMP 192.168.56.1 192.168.56.101 - - 60 - - - - 8 0 - RECEIVE
2013-04-24 10:34:36 DROP UDP 192.168.56.1 192.168.56.255 137 137 78 - - - - - - - RECEIVE
2013-04-24 10:34:45 DROP UDP 192.168.56.1 192.168.56.255 138 138 235 - - - - - - - RECEIVE
```

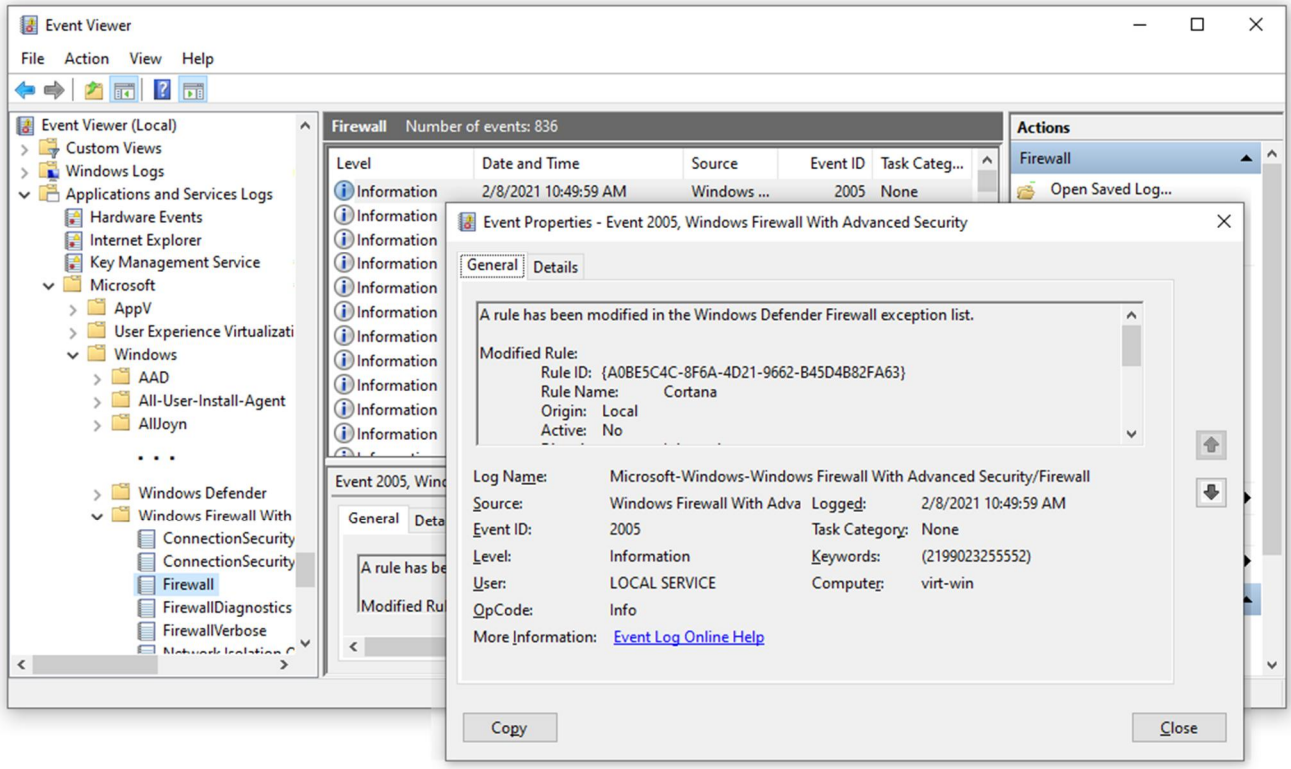
Firewall state monitoring can also be set up by the firewall management applet itself:



It is possible to monitor the current activity of individual rules:



Filtering rules operations monitoring is performed by the system Event Viewer:



🕮 Further reading:

Microsoft support (`netsh advfirewall`) https://support.microsoft.com/kb/947709

🗒 Problems to discuss:

- What does IPC$ share mean, and what is it used for?
- How does remote access to a network share work for a user whose local account (on the system providing the share) does not have a password?
- What do the `net share`, `net view`, `net use`, `net file` (`net files`), and `net session` (`net sessions`) commands allow you to do?
- Review existing filtering rules for inbound traffic in your Windows Firewall. Why do all the predefined rules have an "Allow" action set up?