Task 4 : Setup and Use a Firewall on Windows

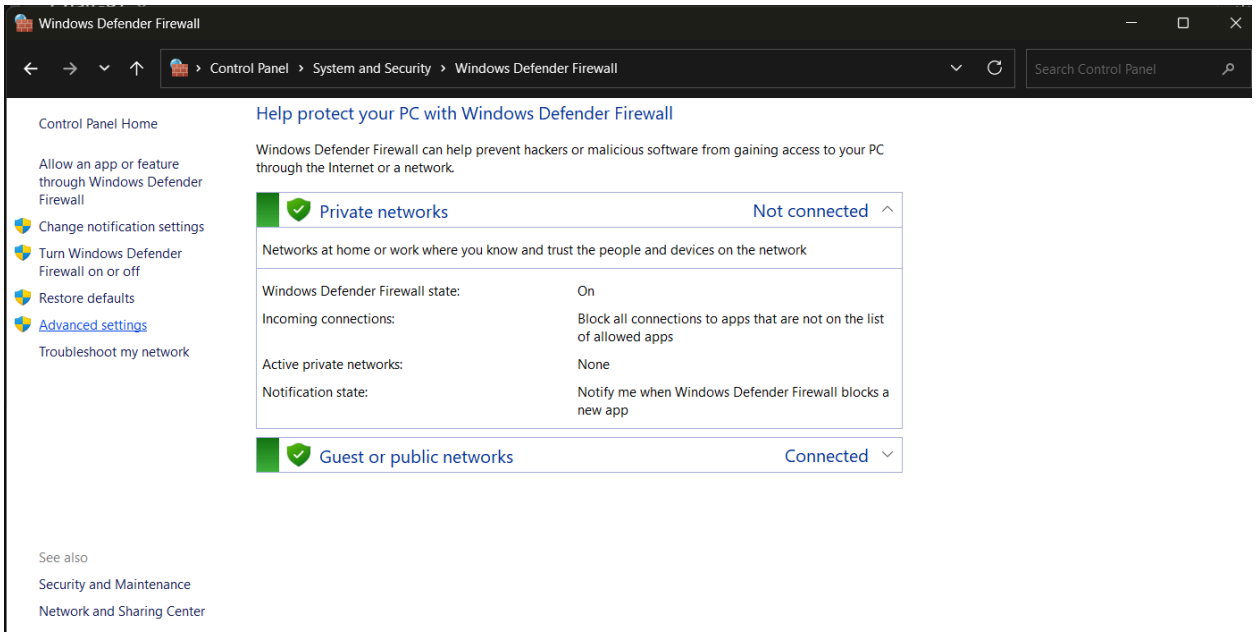
Objective : Setup and Use a Firewall on Windows To configure and test basic firewall rules on Windows that block or allow network traffic, ensuring a deeper understanding of how firewall filtering works

1. Opening the Firewall Configuration Tool

To begin configuring firewall rules, the Windows Defender Firewall with Advanced Security tool was accessed using the graphical user interface.

- Navigated through: Control Panel → System and Security → Windows Defender Firewall → Advanced Settings

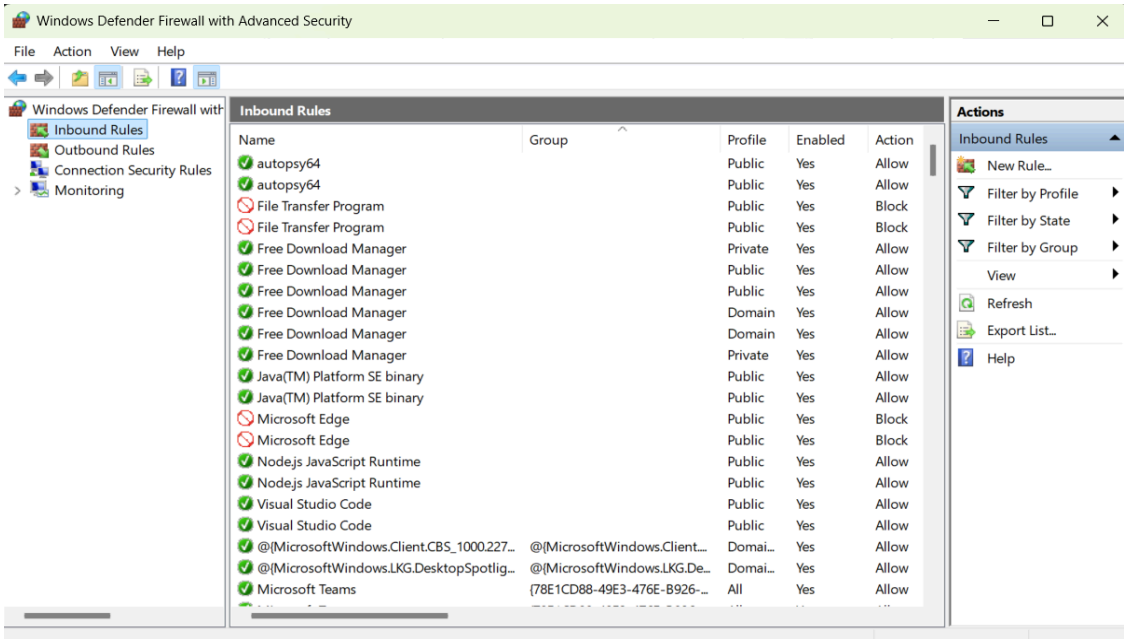
This method opened the Windows Defender Firewall with Advanced Security console, which provides full access to Inbound Rules,Outbound Rules,Connection Security Rules,Monitoring Tools.Using this interface, it is possible to create, edit, enable/disable, or delete custom and system-defined firewall rules.



2. Listing Existing Firewall Rules

After accessing the firewall console, the Inbound Rules and Outbound Rules sections were opened from the left panel. This displayed both system-defined

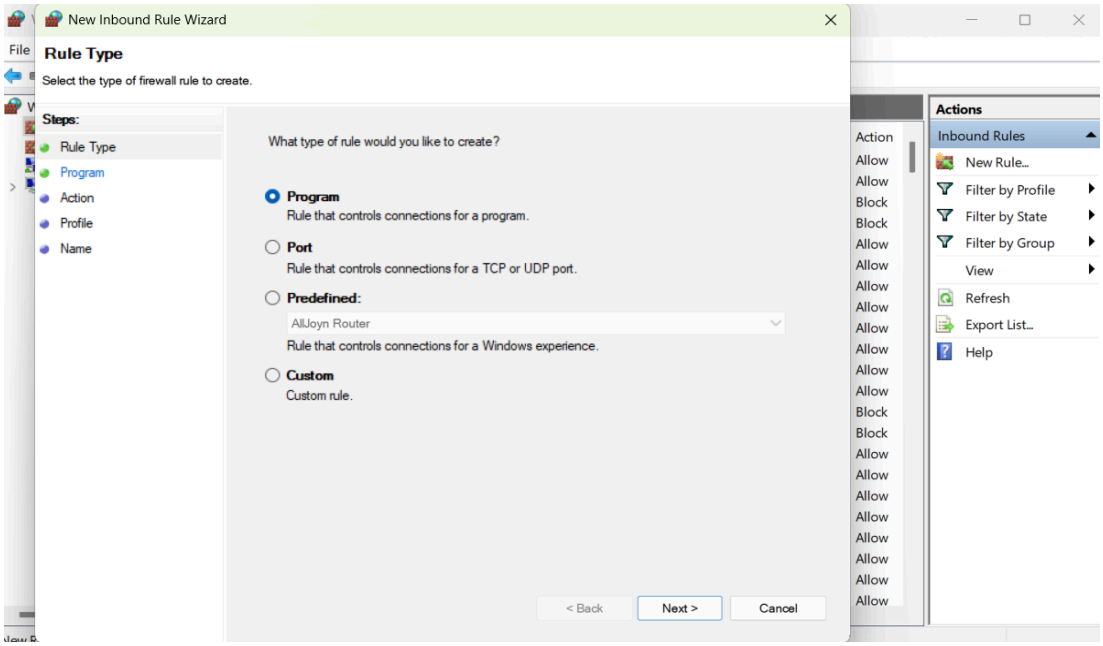
and custom rules, providing a clear view of which ports and services were already managed by the firewall.

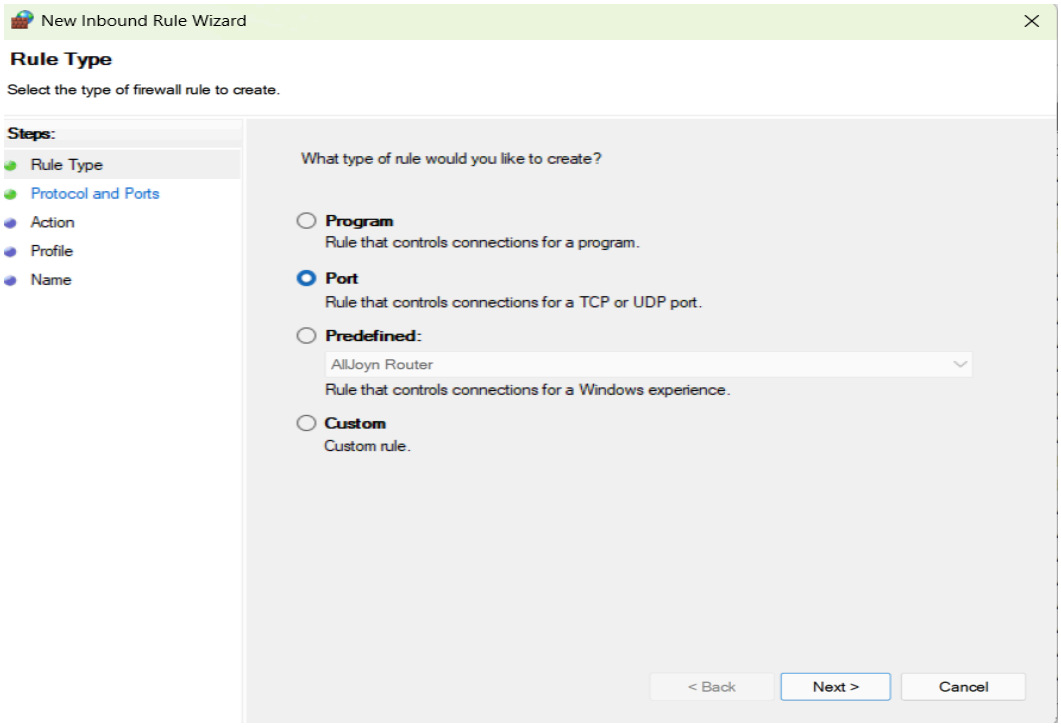


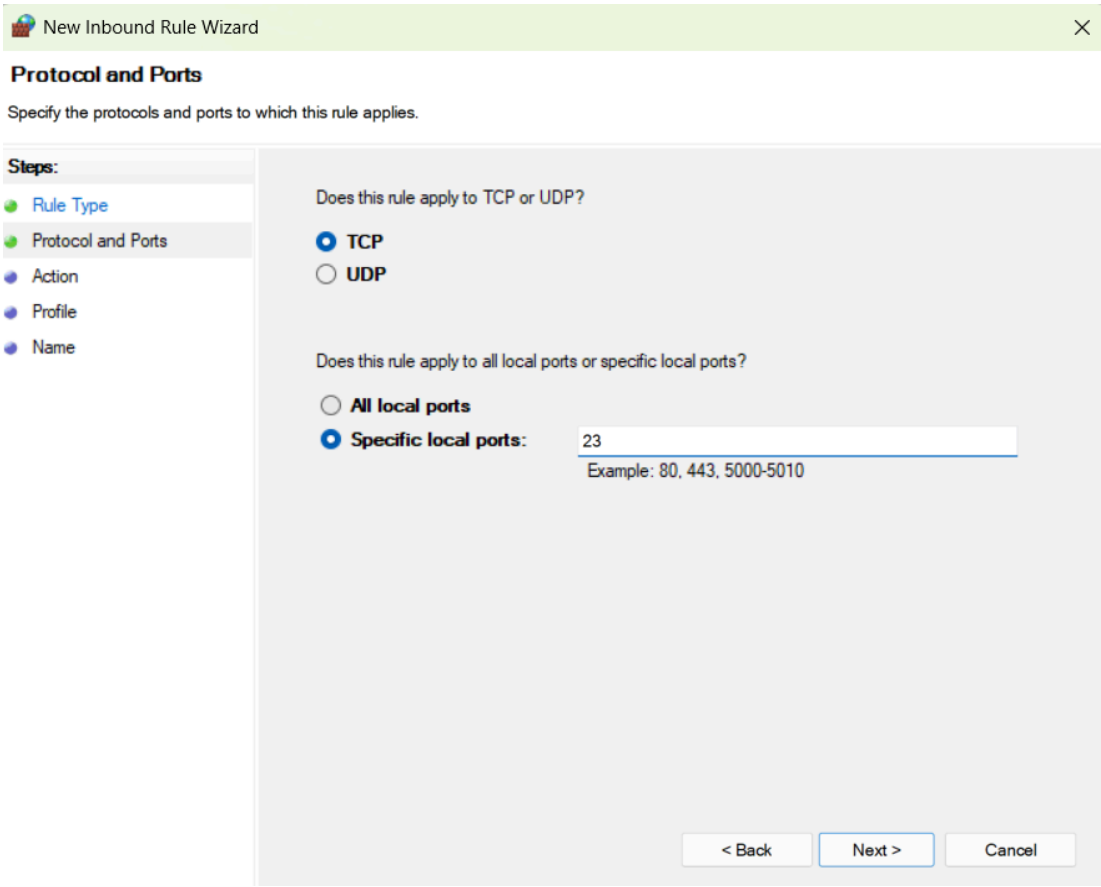
3. Blocking Inbound Traffic on Port 23 (Telnet)

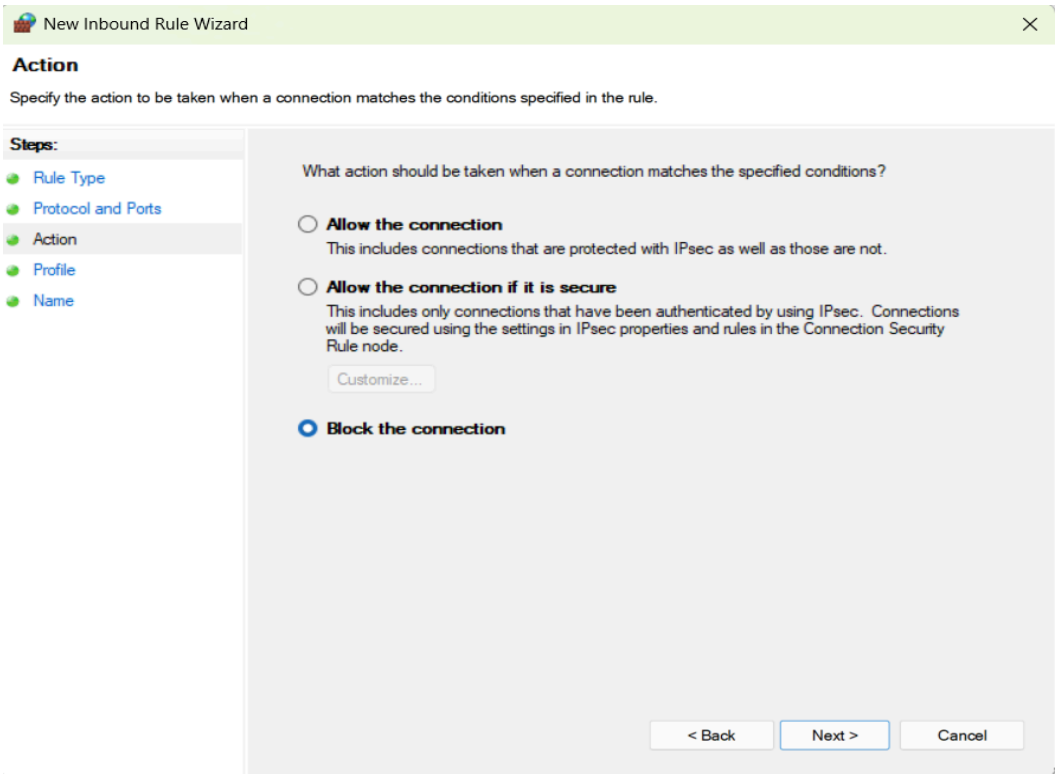
To simulate a basic security policy:

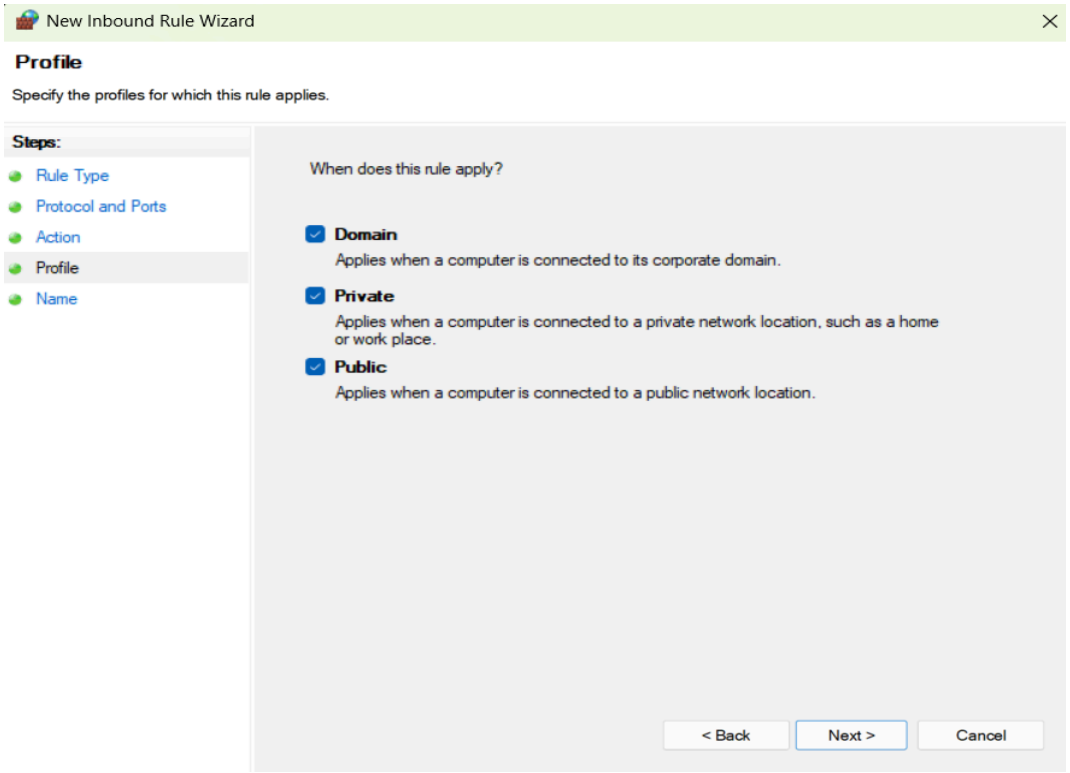
- Steps:
  Inbound Rules → New Rule → Port → TCP → Port 23 → Block → All profiles
  Named: Block Telnet Port 23
  This blocked all inbound Telnet connections.

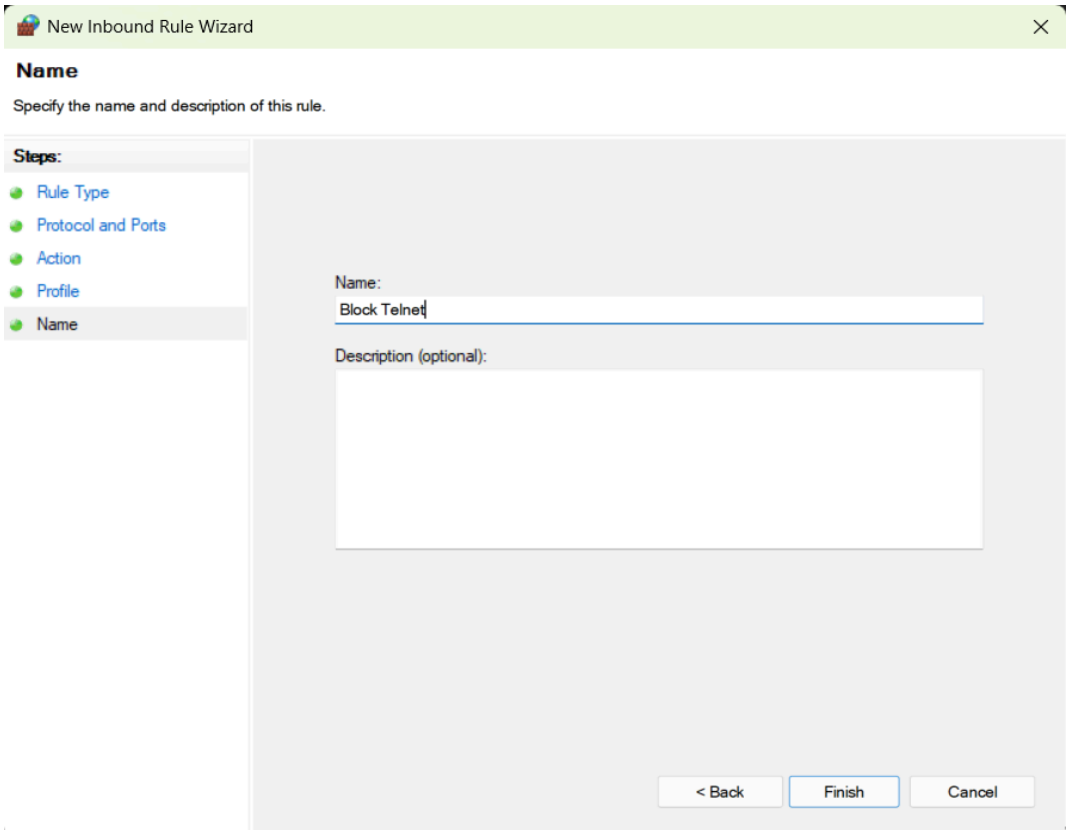

New Inbound Rule Wizard
Rule Type
Select the type of firewall rule to create.
Steps:
Rule Type
Program
Action
Profile
Name
What type of rule would you like to create?
Program
Rule that controls connections for a program.
Port
Rule that controls connections for a TCP or UDP port.
Predefined:
AllJoyn Router
Rule that controls connections for a Windows experience.
Custom
Custom rule.
< Back
Next >
Cancel
Actions
Inbound Rules
New Rule...
Filter by Profile
Filter by State
Filter by Group
View
Refresh
Export List...
Help


New Inbound Rule Wizard
Rule Type
Select the type of firewall rule to create.
Steps:
Rule Type
Protocol and Ports
Action
Profile
Name
What type of rule would you like to create?
Program
Rule that controls connections for a program.
Port
Rule that controls connections for a TCP or UDP port.
Predefined:
AllJoyn Router
Rule that controls connections for a Windows experience.
Custom
Custom rule.
< Back
Next >
Cancel


New Inbound Rule Wizard
Protocol and Ports
Specify the protocols and ports to which this rule applies.
Steps:
Rule Type
Protocol and Ports
Action
Profile
Name
Does this rule apply to TCP or UDP?
TCP
UDP
Does this rule apply to all local ports or specific local ports?
All local ports
Specific local ports:
23
Example: 80, 443, 5000-5010
< Back
Next >
Cancel


New Inbound Rule Wizard
Action
Specify the action to be taken when a connection matches the conditions specified in the rule.
Steps:
Rule Type
Protocol and Ports
Action
Profile
Name
What action should be taken when a connection matches the specified conditions?
Allow the connection
This includes connections that are protected with IPsec as well as those are not.
Allow the connection if it is secure
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.
Customize...
Block the connection
< Back
Next >
Cancel


New Inbound Rule Wizard
Profile
Specify the profiles for which this rule applies.
Steps:
Rule Type
Protocol and Ports
Action
Profile
Name
When does this rule apply?
Domain
Applies when a computer is connected to its corporate domain.
Private
Applies when a computer is connected to a private network location, such as a home or work place.
Public
Applies when a computer is connected to a public network location.
< Back
Next >
Cancel


New Inbound Rule Wizard
Name
Specify the name and description of this rule.
Steps:
Rule Type
Protocol and Ports
Action
Profile
Name
Name:
Block Telnet
Description (optional):
< Back
Finish
Cancel

4. Testing the Telnet Rule

Used PowerShell: Test-NetConnection -ComputerName 127.0.0.1 -Port 23

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\jenna> Test-NetConnection -ComputerName 127.0.0.1 -Port 23
WARNING: TCP connect to (127.0.0.1 : 23) failed


ComputerName            : 127.0.0.1
RemoteAddress           : 127.0.0.1
RemotePort              : 23
InterfaceAlias          : Loopback Pseudo-Interface 1
SourceAddress           : 127.0.0.1
PingSucceeded           : True
PingReplyDetails (RTT)  : 0 ms
TcpTestSucceeded        : False



PS C:\Users\jenna>
```

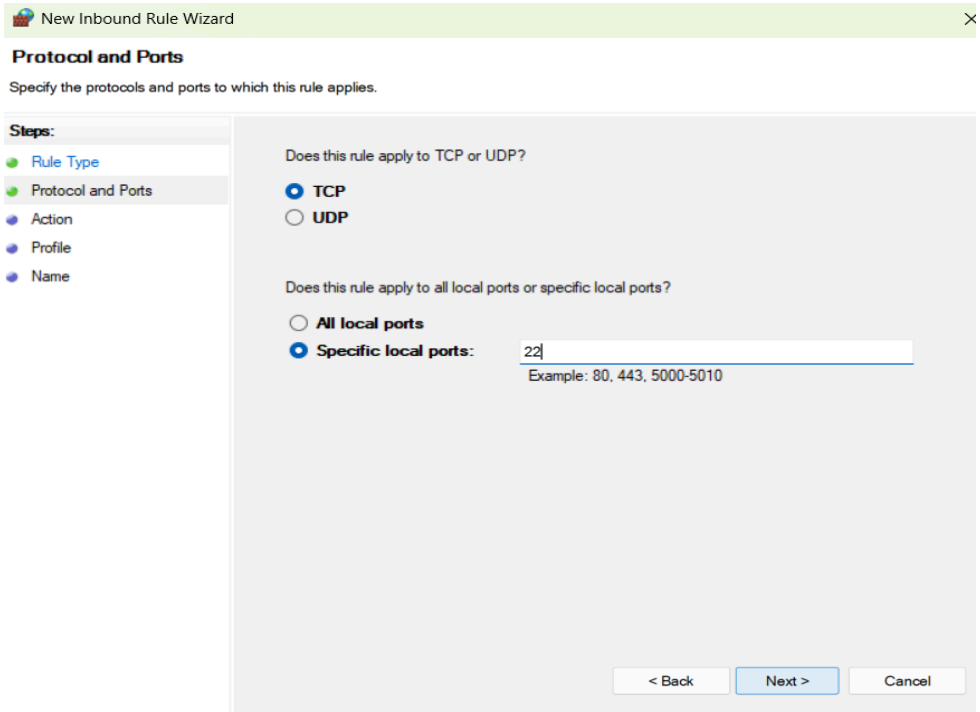
5. Allowing SSH (Port 22)

Created a rule to allow SSH:

Steps:

Inbound Rules → New Rule → Port → TCP → Port 22 → Allow → All profiles

Named: Allow SSH Port 22



New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

What action should be taken when a connection matches the specified conditions?

Allow the connection
This includes connections that are protected with IPsec as well as those are not.

Allow the connection if it is secure
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

Customize...

Block the connection

< Back | Next > | Cancel

New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

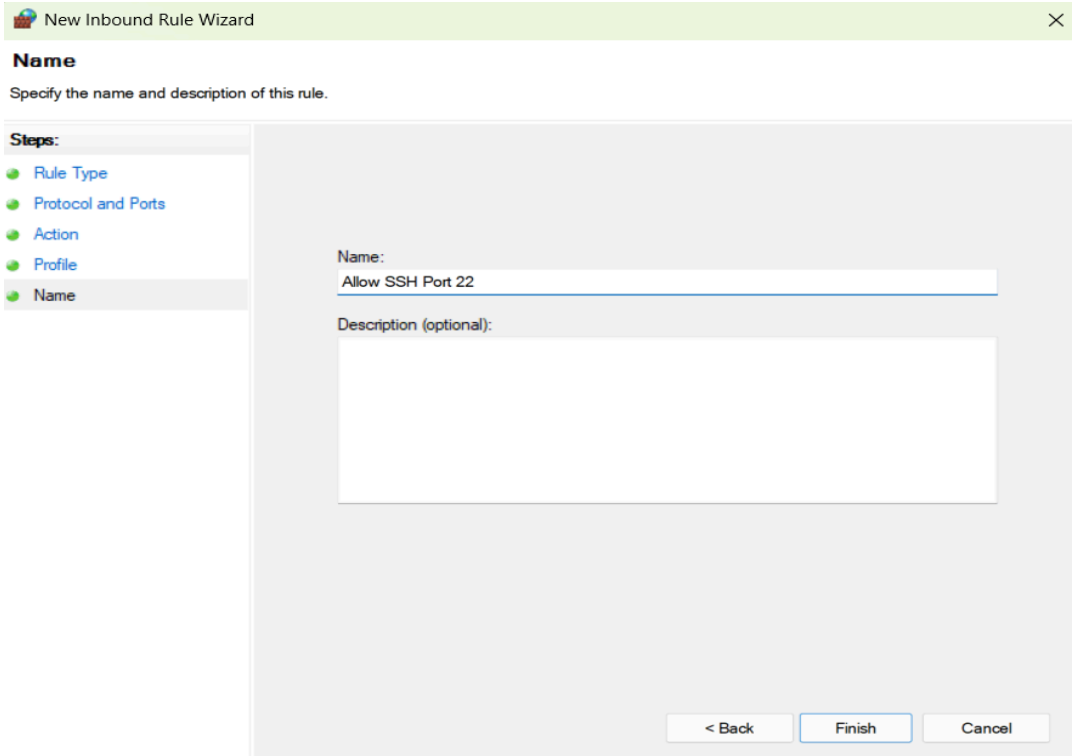
When does this rule apply?

Domain
Applies when a computer is connected to its corporate domain.

Private
Applies when a computer is connected to a private network location, such as a home or work place.

Public
Applies when a computer is connected to a public network location.

< Back | Next > | Cancel




```
PS C:\Users\jenna> Test-NetConnection -ComputerName 127.0.0.1 -Port 22
WARNING: TCP connect to (127.0.0.1 : 22) failed


ComputerName            : 127.0.0.1
RemoteAddress           : 127.0.0.1
RemotePort              : 22
InterfaceAlias          : Loopback Pseudo-Interface 1
SourceAddress           : 127.0.0.1
PingSucceeded           : True
PingReplyDetails (RTT)  : 0 ms
TcpTestSucceeded        : False
```

6. Removing the Telnet Rule

To restore default state:

- Located "Block Telnet Port 23" under Inbound Rules → Right-click → Delete.

7. Manual Port Listener Test

To simulate a listening service:

```
$listener = [System.Net.Sockets.TcpListener]23
$listener.Start()
```

- Even with a listener active, `Test-NetConnection` failed if the firewall block rule was active — confirming the firewall was working.

To stop the listener: $listener.Stop()

8. Real-World Test Results with Test-NetConnection

<u>Test 1: SSH on localhost (Port 22)</u>

Test-NetConnection -ComputerName 127.0.0.1 -Port 22

```
PS C:\Users\jenna> Test-NetConnection -ComputerName 127.0.0.1 -Port 22
WARNING: TCP connect to (127.0.0.1 : 22) failed


ComputerName            : 127.0.0.1
RemoteAddress           : 127.0.0.1
RemotePort              : 22
InterfaceAlias          : Loopback Pseudo-Interface 1
SourceAddress           : 127.0.0.1
PingSucceeded           : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded        : False
```

No SSH service was running; firewall allowed it, but no connection possible without a listener.

<u>Test 2: Telnet on remote IP (initially failed)</u>

Test-NetConnection -ComputerName 192.168.1.9 -Port 23

```
PS C:\Users\jenna> Test-NetConnection -ComputerName 192.168.1.9 -Port 23
WARNING: TCP connect to (192.168.1.9 : 23) failed
WARNING: Ping to 192.168.1.9 failed with status: TimedOut


ComputerName            : 192.168.1.9
RemoteAddress           : 192.168.1.9
RemotePort              : 23
InterfaceAlias          : Wi-Fi
SourceAddress           : 192.168.1.2
PingSucceeded           : False
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded        : False
```

Target was unreachable, likely off or blocked.

Test 3: Telnet on same remote IP (later succeeded)

Test-NetConnection -ComputerName 192.168.1.9 -Port 23

```
PS C:\Users\jenna> Test-NetConnection -ComputerName 192.168.1.9 -Port 23


ComputerName     : 192.168.1.9
RemoteAddress    : 192.168.1.9
RemotePort       : 23
InterfaceAlias   : Wi-Fi
SourceAddress    : 192.168.1.2
TcpTestSucceeded : True
```

Target was now reachable and had port 23 open and listening.