



# Understanding and Managing Windows Firewall (Practical Session)

Gergő Ládi

Laboratory of Cryptography and System Security  
Department of Networked Systems and Services  
Gergo.Ladi@CrySyS.hu



# Outline

---

- Recap (firewalls in general)
- Understanding Windows Firewall
  - A short history
  - Windows Filtering Platform
  - Windows (Defender) Firewall
- Managing Windows Firewall
  - Via GUI
  - Via netsh
  - Via PowerShell

# Recap: What is a firewall?

---

- A firewall is a system (or group thereof) that monitors and/or controls inbound and/or outbound network traffic based on a set of administrator-defined rules (the policy)
  - One of the many definitions
- A typical classification
  - Packet filtering firewall
    - » Makes decisions based on L3/L4 header information
    - » Does not track connection states
  - Stateful firewall
    - » Like a packet filtering firewall, but keeps track of connection states
  - Application-layer firewall
    - » Can understand (some) L7 protocols and make decisions based on L7 contents

A SHORT HISTORY

---

# **UNDERSTANDING WINDOWS FIREWALL**

# Windows Firewall – A short history

---

- Windows 98 ME and earlier editions
  - No built-in firewall!
  - Users had to find, download, and install one themselves
  - No dedicated Windows APIs for developers
    - » NDIS "hacks", Winsock LSPs, TDI Filter Drivers were necessary

# Windows Firewall – A short history

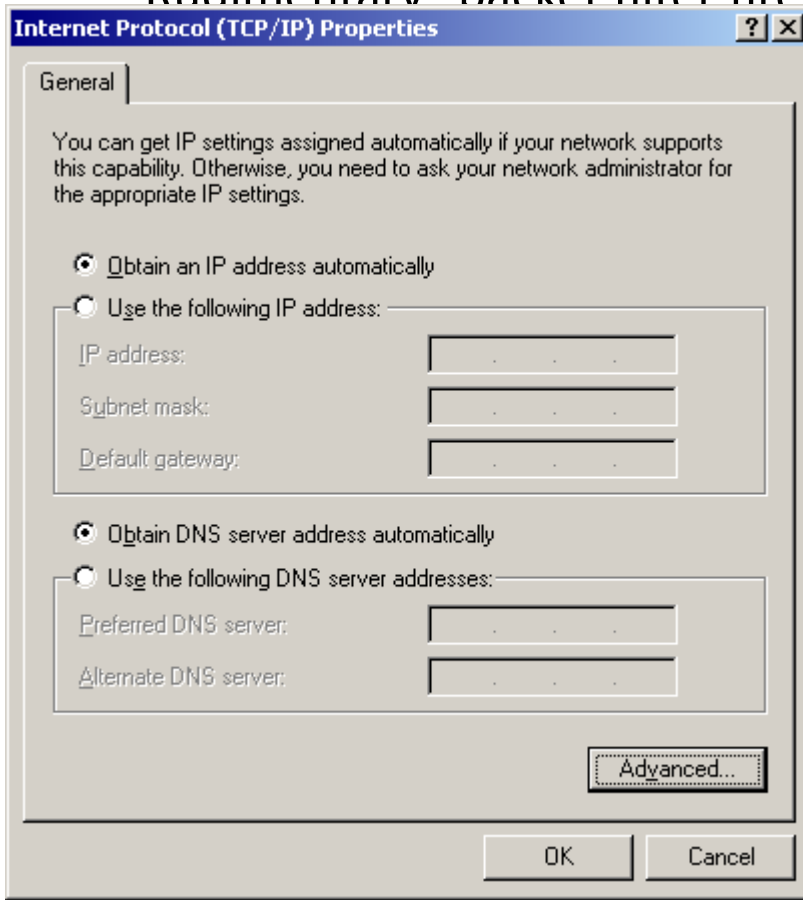
---

- Windows 2000 and Windows XP until SP2
  - Rudimentary "packet filter firewall"
    - » Can filter only incoming traffic
    - » Can filter only TCP, UDP, and IP by 'Protocol' header
  - Some support for firewall developers
    - » Filter Hook Driver (one per system), Firewall Hook Driver

# Windows Firewall – A short history

- Windows 2000 and Windows XP until SP2

– Rudimentary "packet filter firewall"



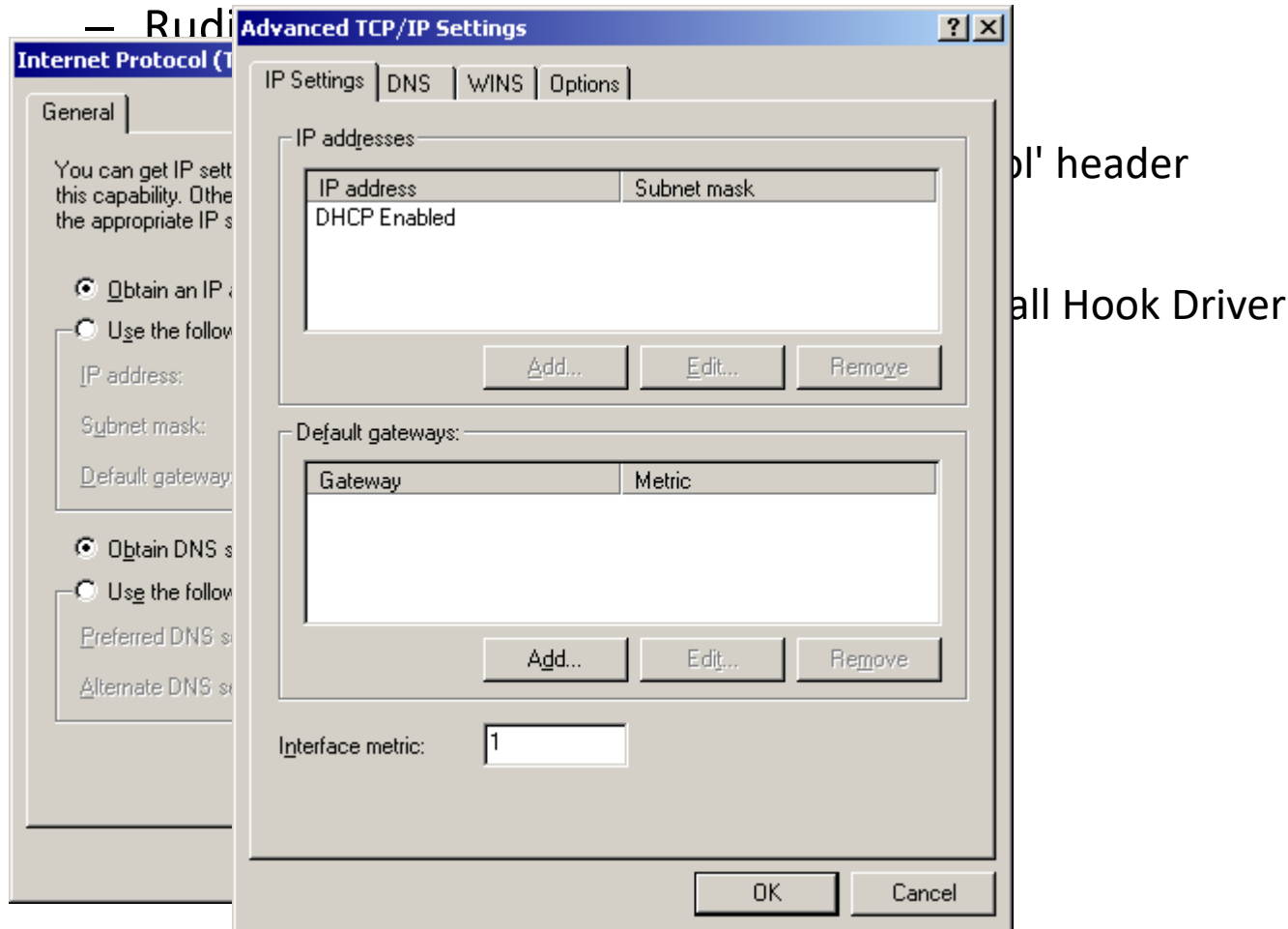
by 'Protocol' header

lopers

(system), Firewall Hook Driver

# Windows Firewall – A short history

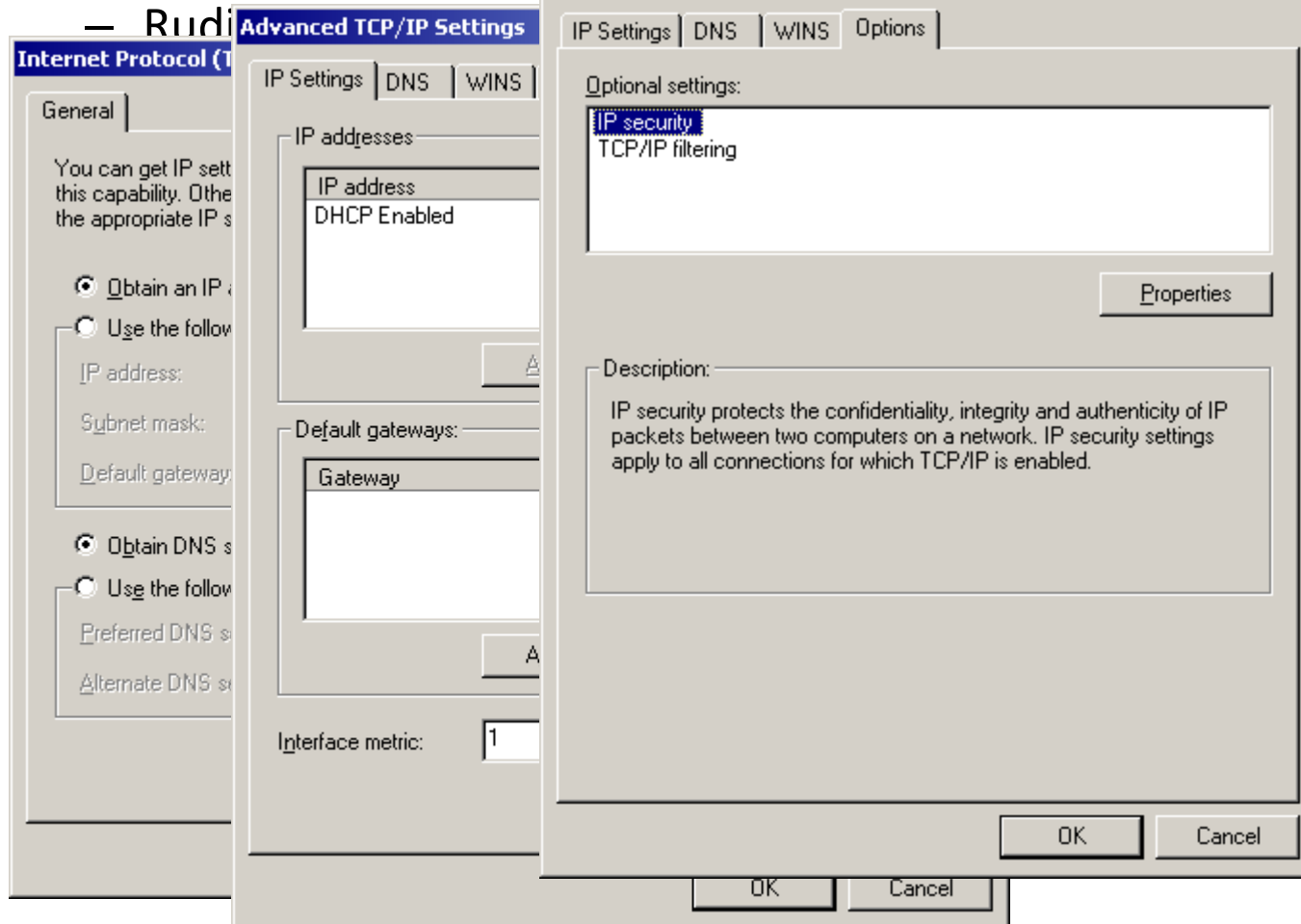
- Windows 2000 and Windows XP until SP2





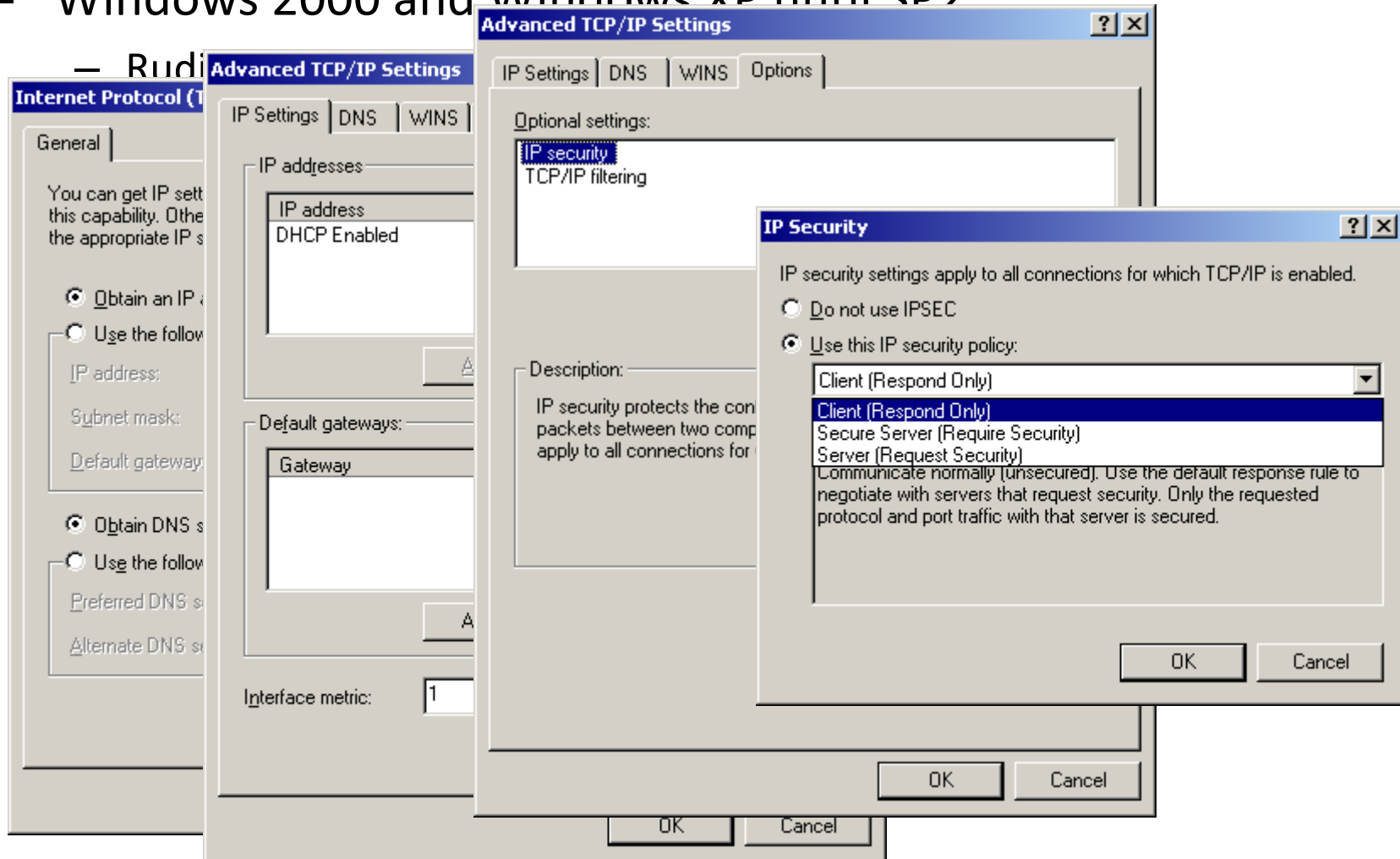
# Windows Firewall – A short history

- Windows 2000 and Windows XP until SP2



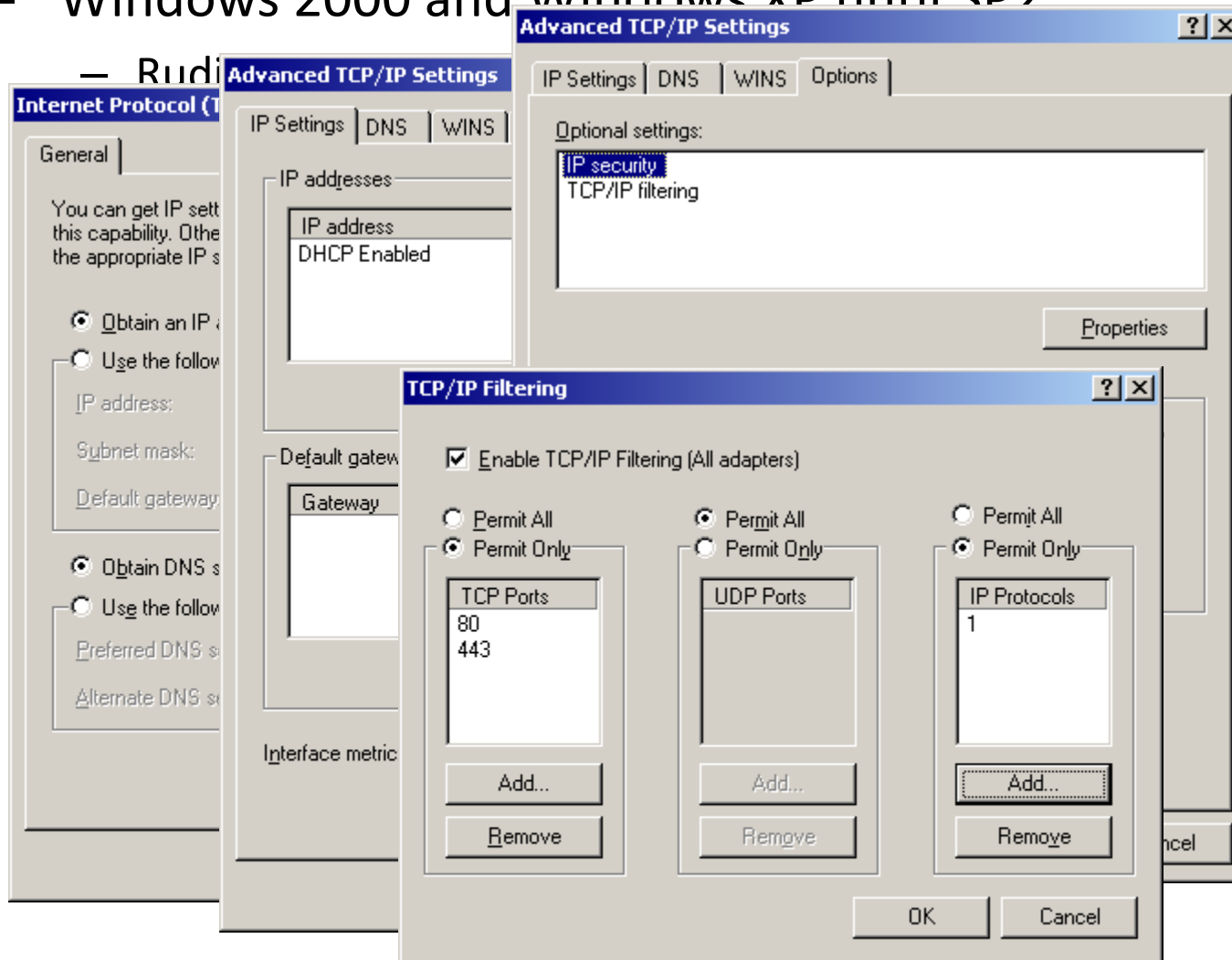
# Windows Firewall – A short history

- Windows 2000 and Windows XP until SP2



# Windows Firewall – A short history

- Windows 2000 and Windows XP until SP2



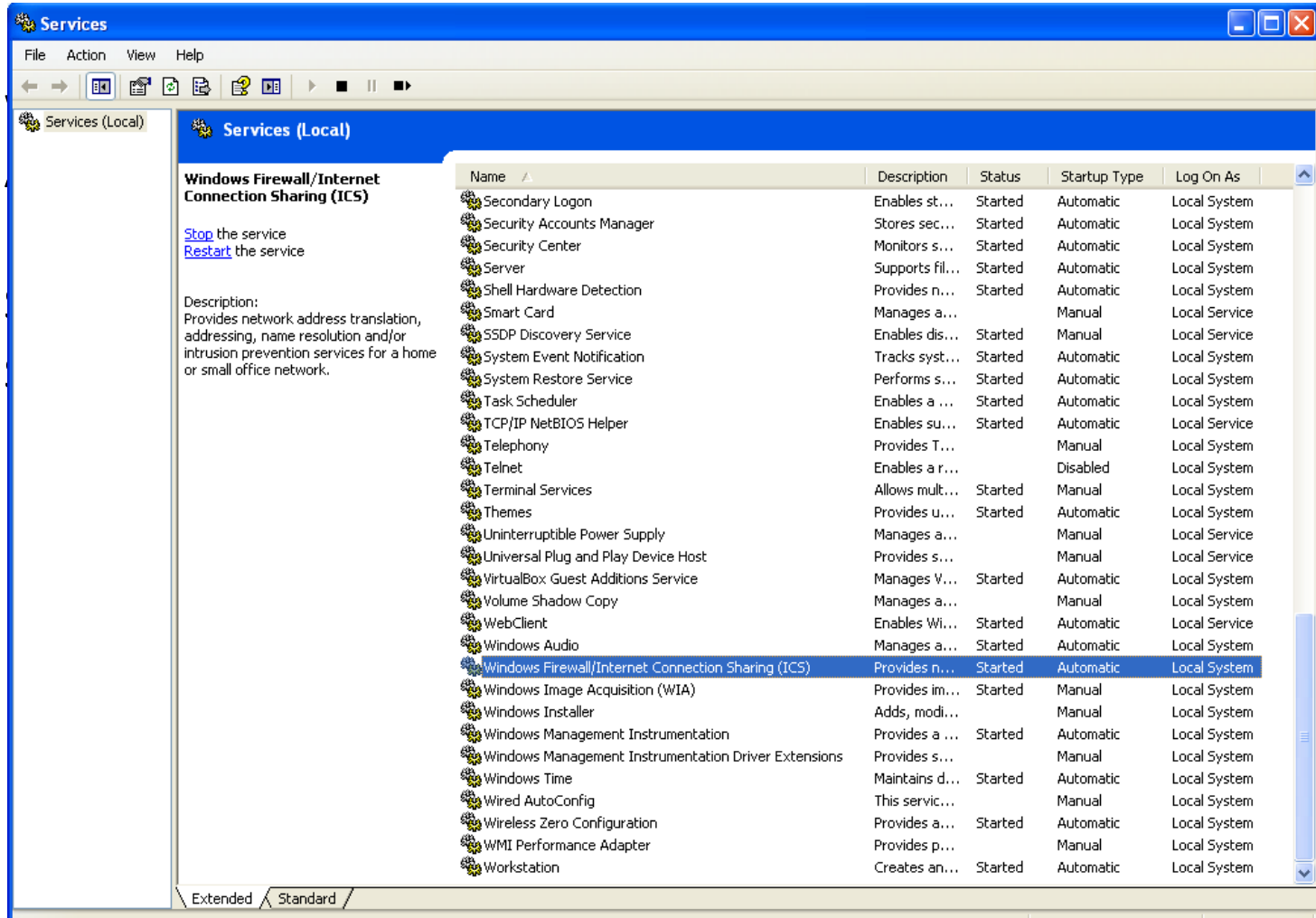
# Windows Firewall – A short history

---

- Windows XP SP2
  - Firewall now runs as a service, and its functionality has been merged with Windows Internet Connection Sharing (ICS), a.k.a. SharedAccess
  - A more user-friendly GUI
    - » The old interface is still available
  - Same developer interface as Windows 2000
  - Some more features were added (compared to Windows 2000)
    - » Applications can be whitelisted
    - » Rule groups
    - » Rule scoping
    - » Logging
    - » Filtering ICMP traffic based on ICMP type
    - » NAT (was a stand-alone service in Windows 2000)

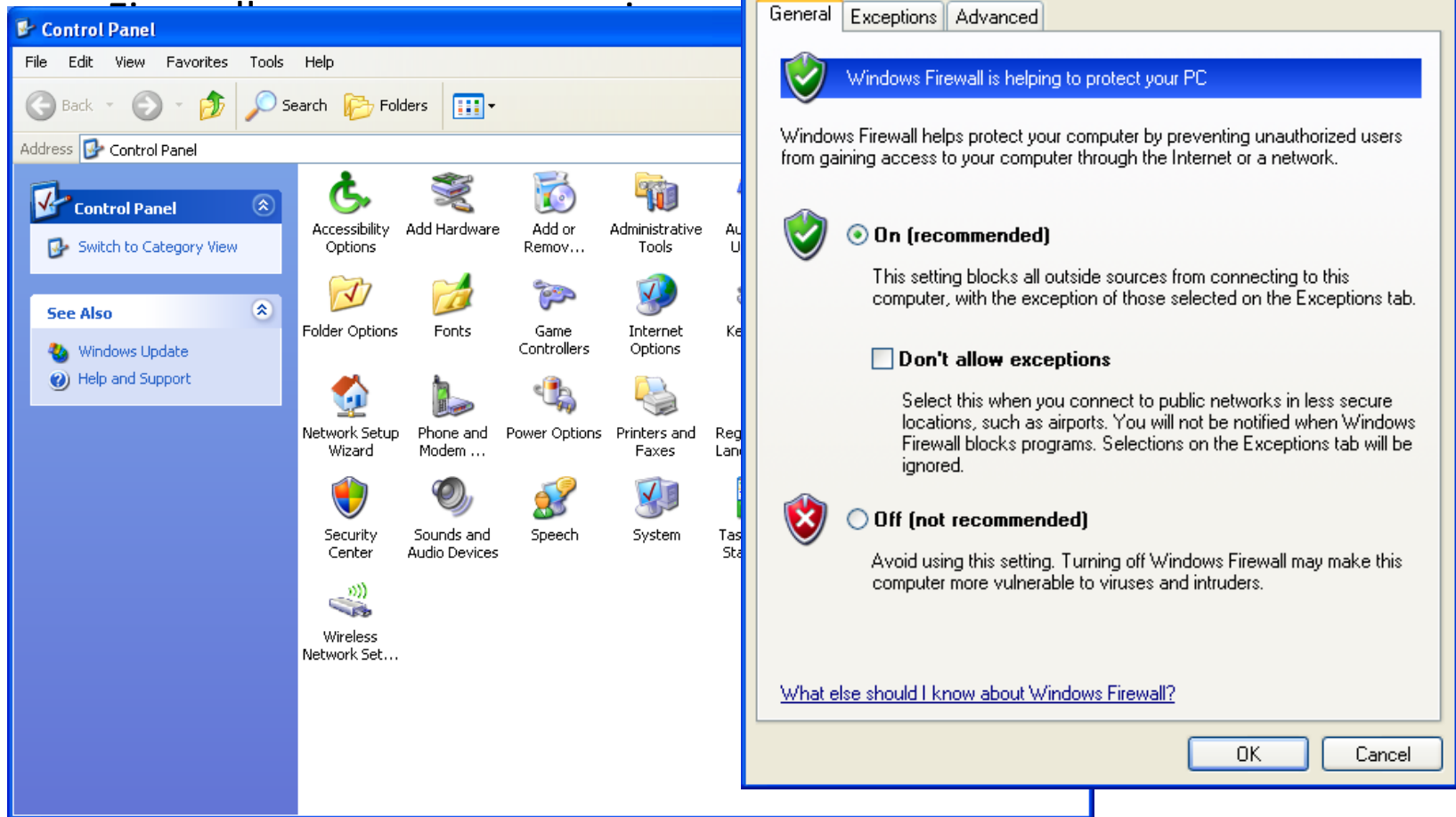
# Windows Firewall – A short history

- Windows XP SP2



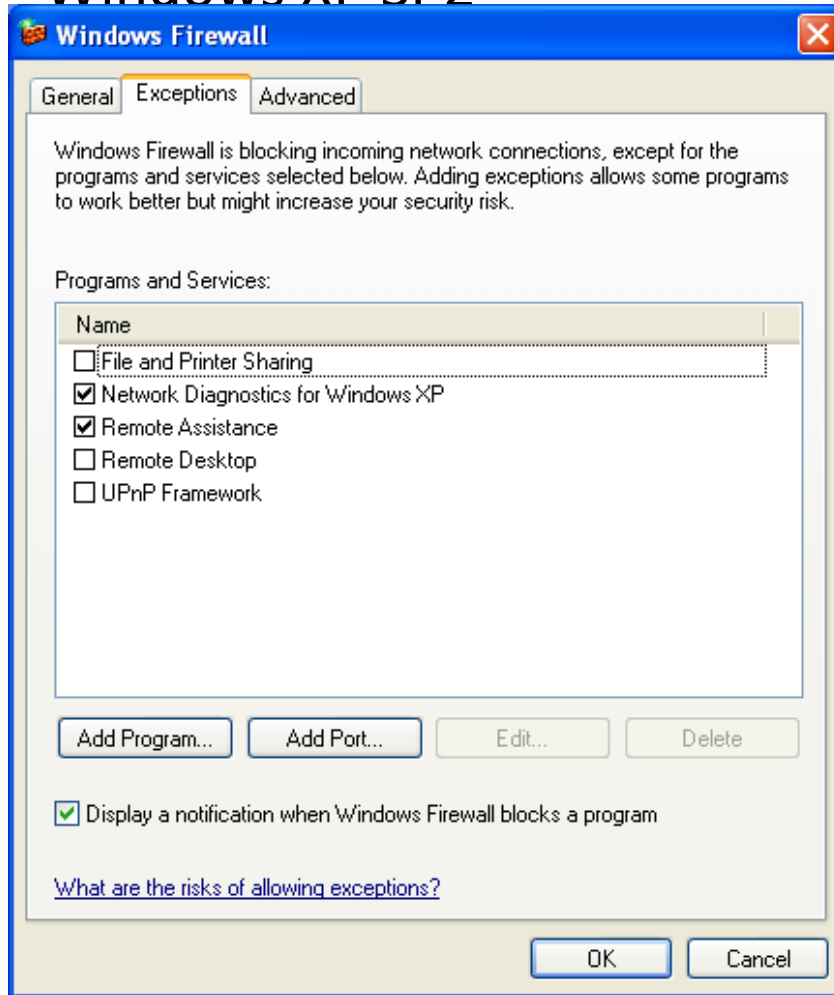
# Windows Firewall – A short history

- Windows XP SP2

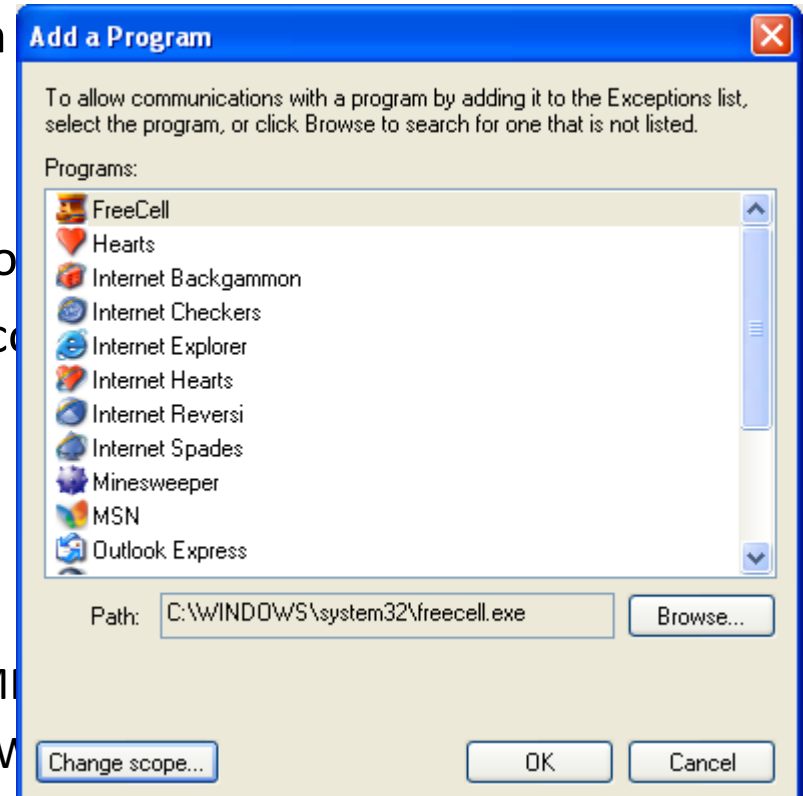


# Windows Firewall – A short history

## ■ Windows XP SP2

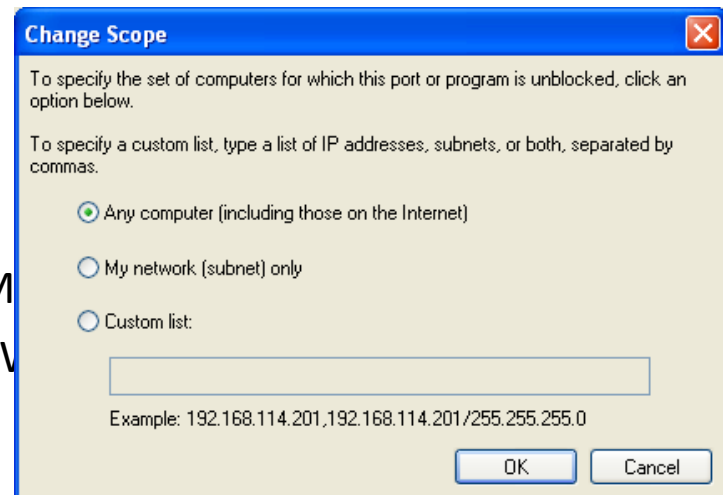
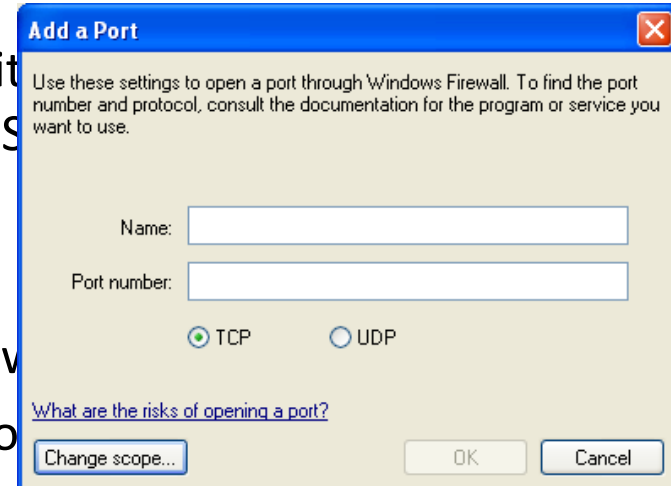
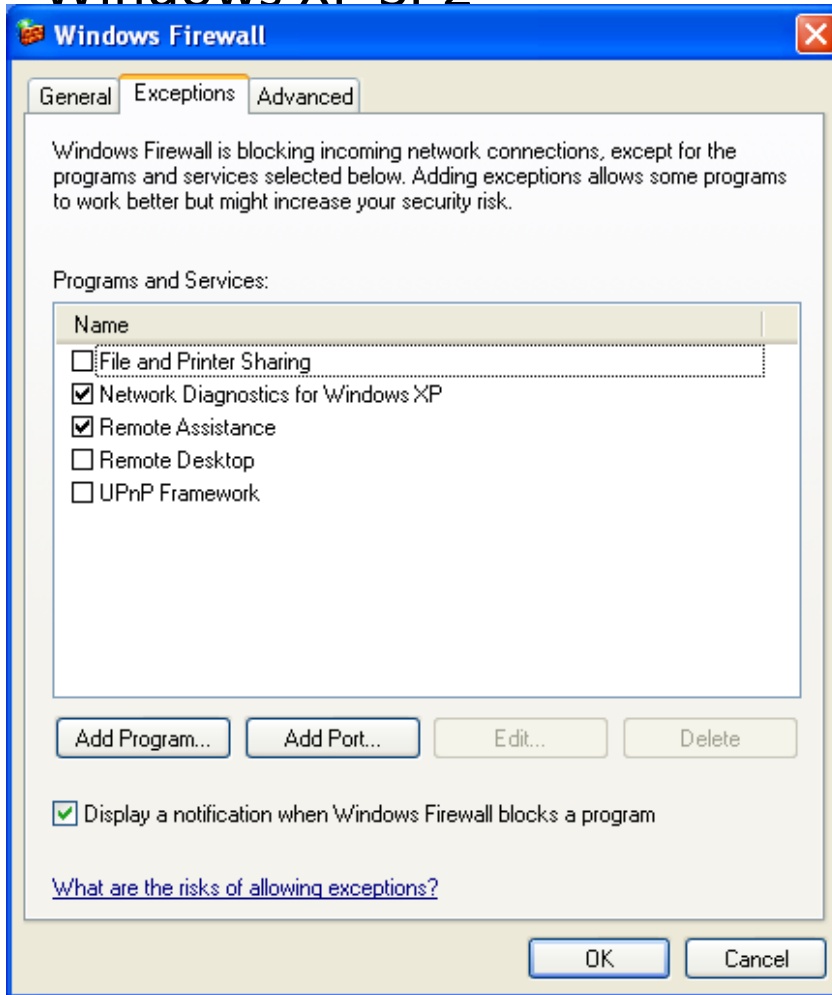


and its functionality has been merged



# Windows Firewall – A short history

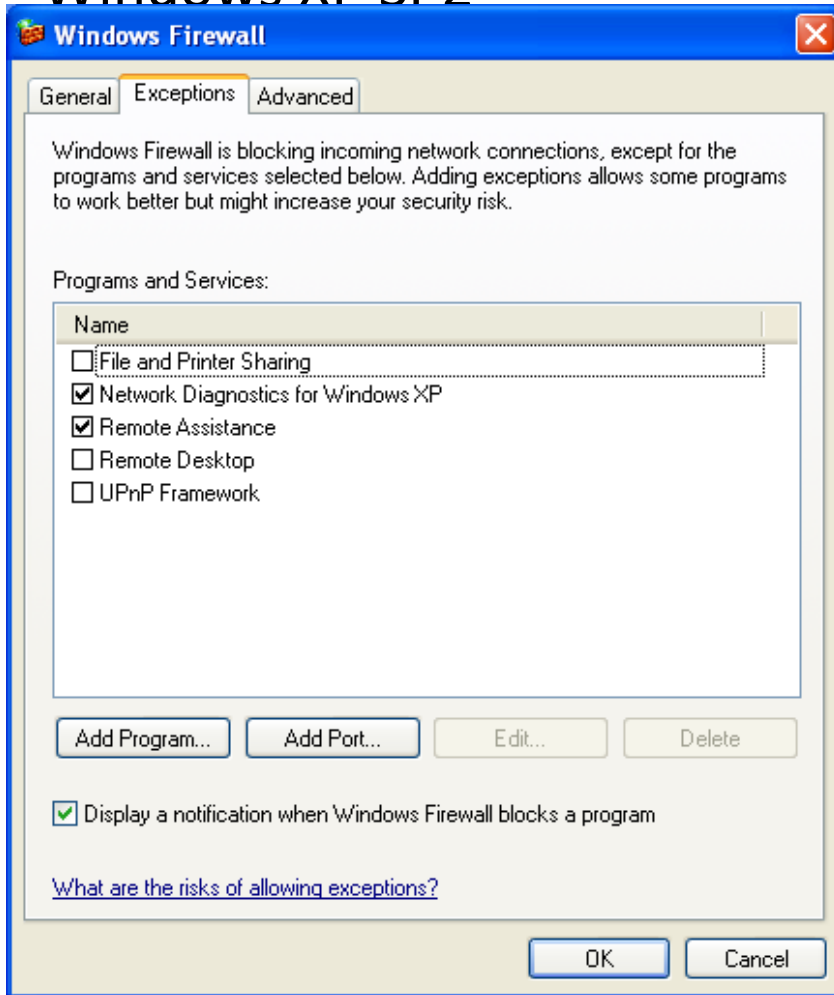
## ■ Windows XP SP2





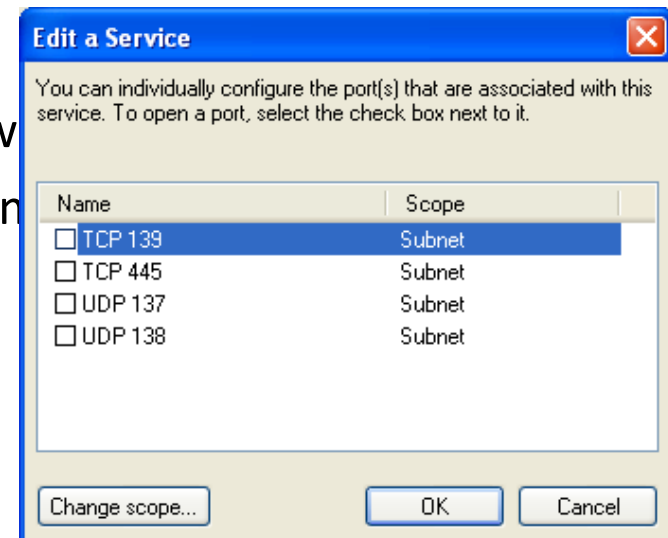
# Windows Firewall – A short history

## ■ Windows XP SP2



and its functionality has been merged  
on Sharing (ICS), a.k.a. SharedAccess

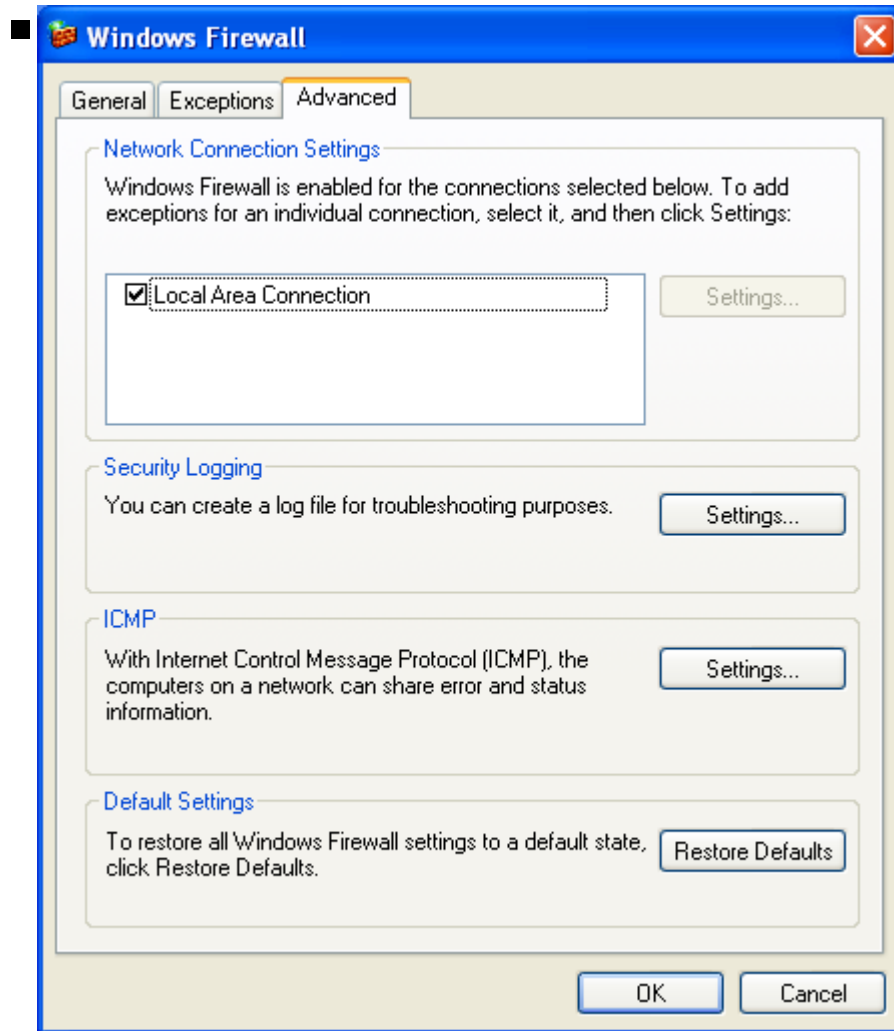
down  
(con



MP type

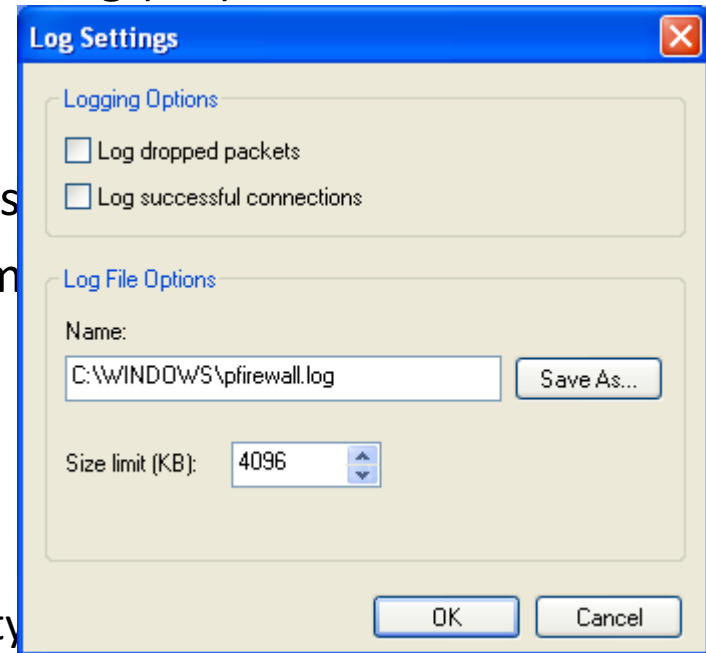
in Windows 2000)

# Windows Firewall – A short history



its functionality has been merged  
n Sharing (ICS), a.k.a. SharedAccess

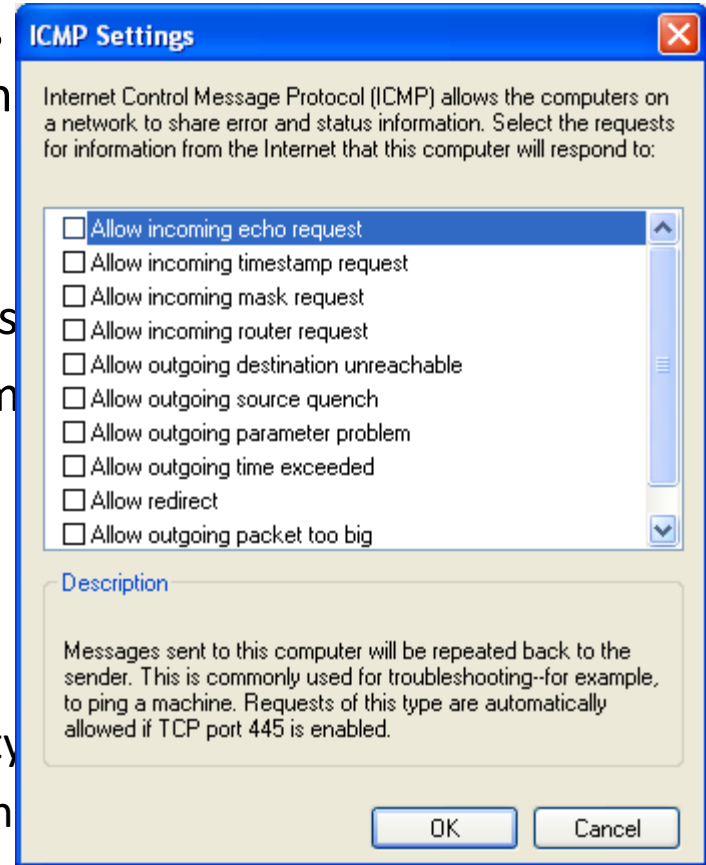
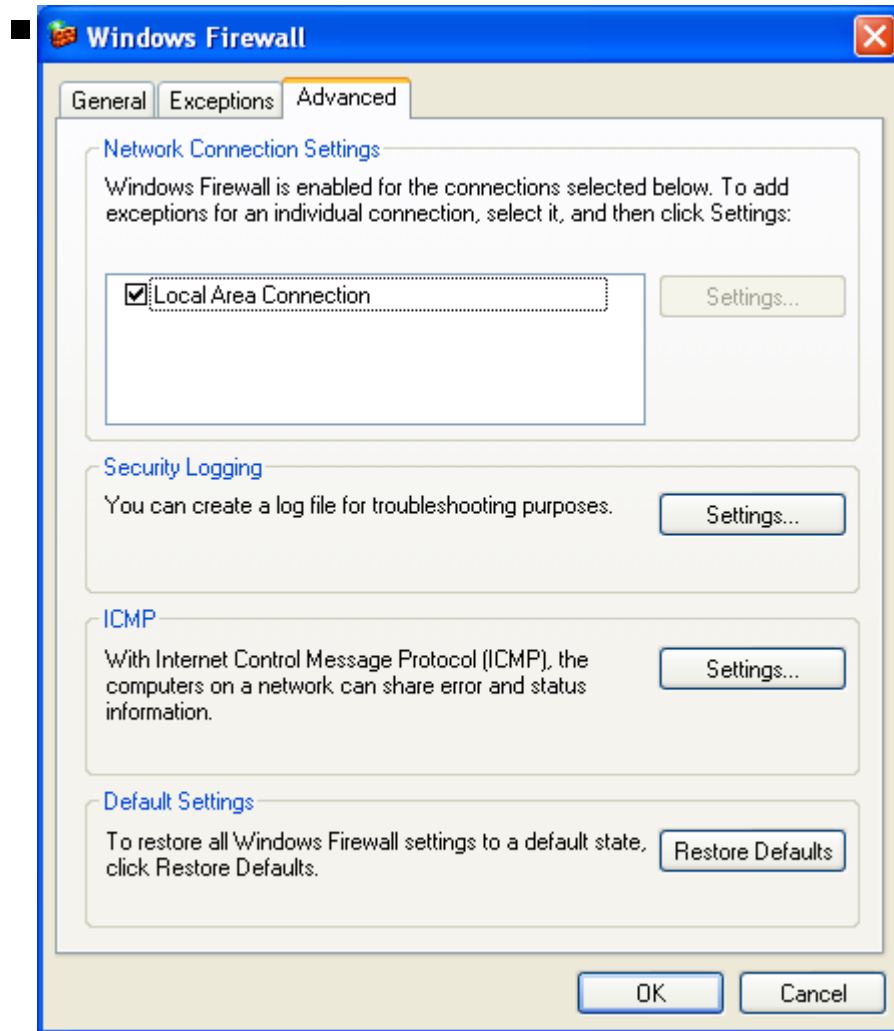
OWS  
com



IP ty

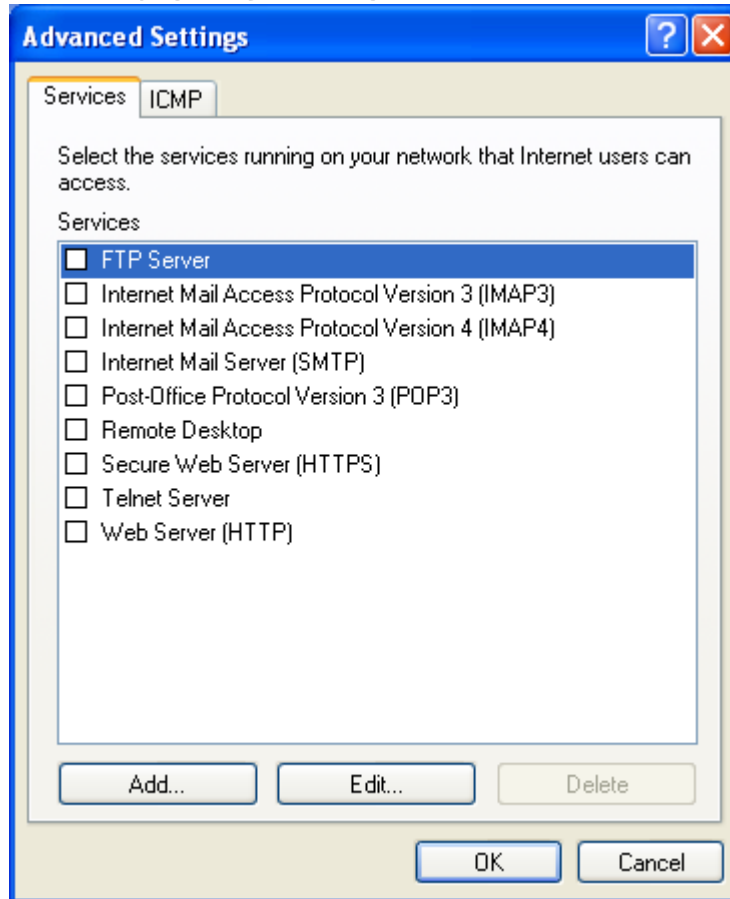
Windows 2000)

# Windows Firewall – A short history

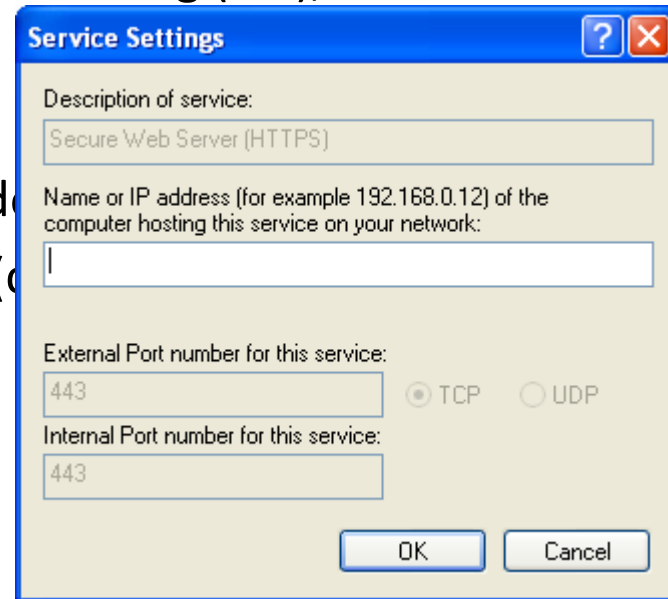


# Windows Firewall – A short history

- Windows XP SP2



, and its functionality has been merged  
ection Sharing (ICS), a.k.a. SharedAccess



able  
Wind  
ded (c  
ed

on ICMP type

ce in Windows 2000)

# Windows Firewall – A short history

---

## ■ Windows XP SP2

### – Basic programmability via netsh firewall

```
C:\Documents and Settings\Admin>netsh firewall
```

The following commands are available:

Commands in this context:

?	- Displays a list of commands.
add	- Adds firewall configuration.
delete	- Deletes firewall configuration.
dump	- Displays a configuration script.
help	- Displays a list of commands.
reset	- Resets firewall configuration to default.
set	- Sets firewall configuration.
show	- Shows firewall configuration.

To view help for a command, type the command, followed by a space, and then type ?.

```
C:\Documents and Settings\Admin>netsh firewall add
```

The following commands are available:

Commands in this context:

add allowedprogram	- Adds firewall allowed program configuration.
add portopening	- Adds firewall port configuration.

THE PRESENT

---

# **UNDERSTANDING WINDOWS FIREWALL**

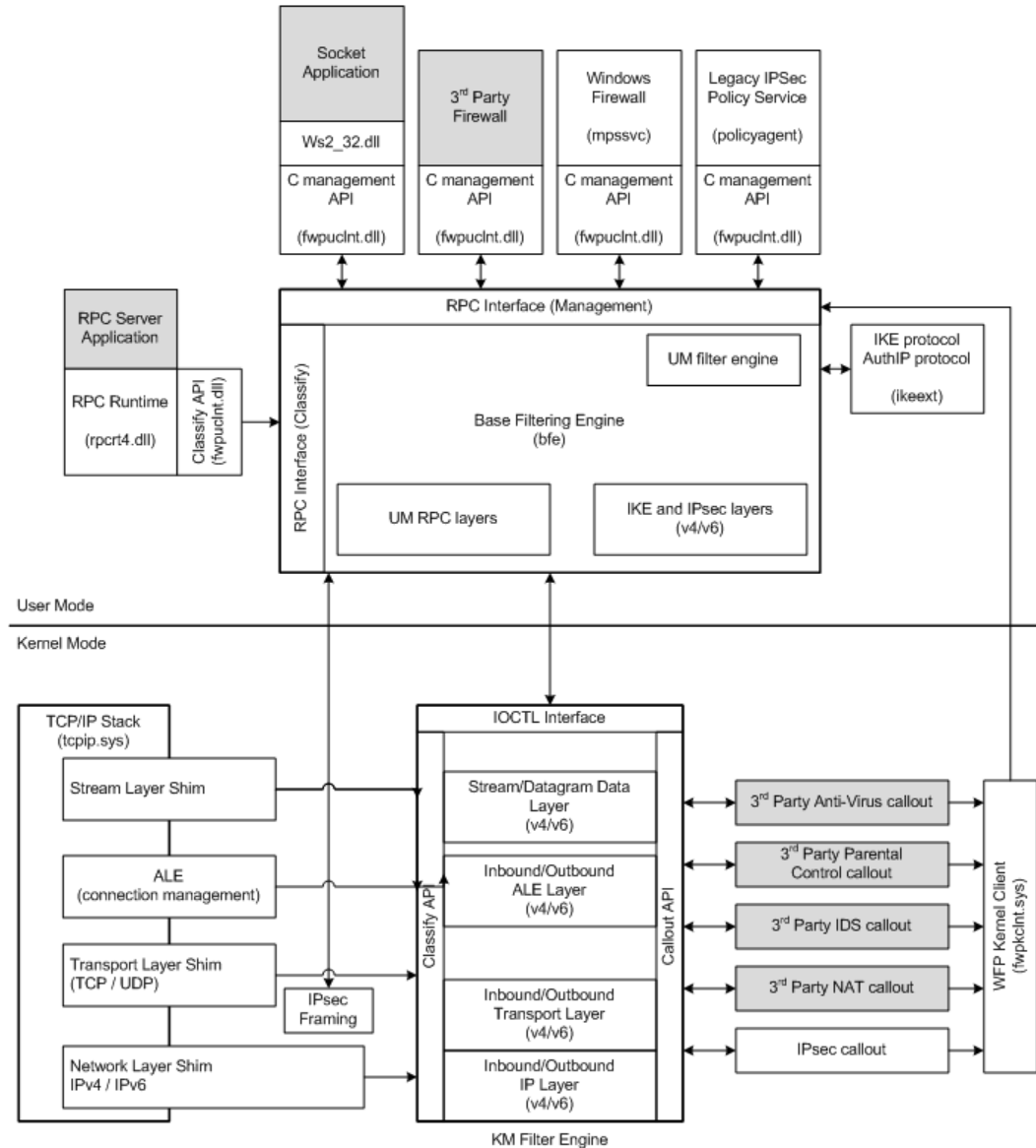
# Windows Filtering Platform (WFP)

---

- Introduced in Windows Vista (and is still used today)
- Completely new architecture for packet filtering/manipulation
  - Supersedes all previous filtering methods
    - » Though many are (were) left available for backwards compatibility
  - Designed to be (more) future-proof
  - Modular and developer-friendly
- Implements state tracking for connections -> stateful firewall
- It is now possible to filter outgoing traffic
- Boot-time security
  - Boot-time filters are applied when tcpip.sys is loaded (kernel-level)
  - These filters drop everything except for DHCP and DNS (and Kerberos/SMB in case of domain-joined computers)
  - (It is possible to install additional boot-time filters if needed)

# Windows Filtering Platform (WFP)

- Filter Engine
  - Two components:
    - » User-mode
    - » Kernel-mode
- Base Filtering Engine
  - User-mode component
  - Disables boot-time filters once started
  - Loads/unloads persistent and run-time filters
  - Stores filter configuration



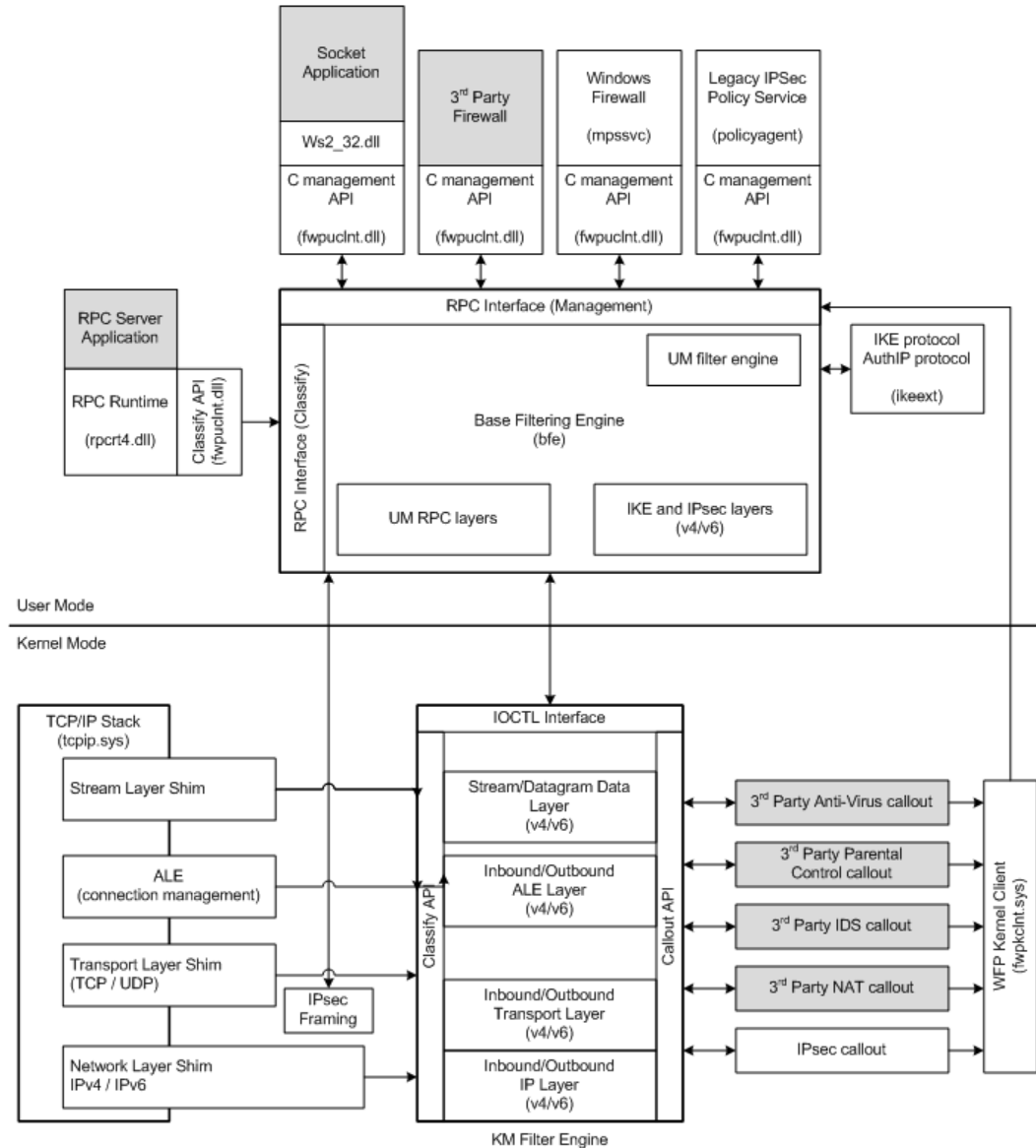
Source: <https://docs.microsoft.com/en-us/windows/desktop/fwp/windows-filtering-platform-architecture-overview>



# Windows Filtering Platform (WFP)

## ■ Shims

- Extract information from incoming/outgoing packets
- Perform classification using the Filter Engine
  - » Filter arbitration (later...)
- Handle packets according to classification results
  - » Permit
  - » Block
  - » Pass to callout



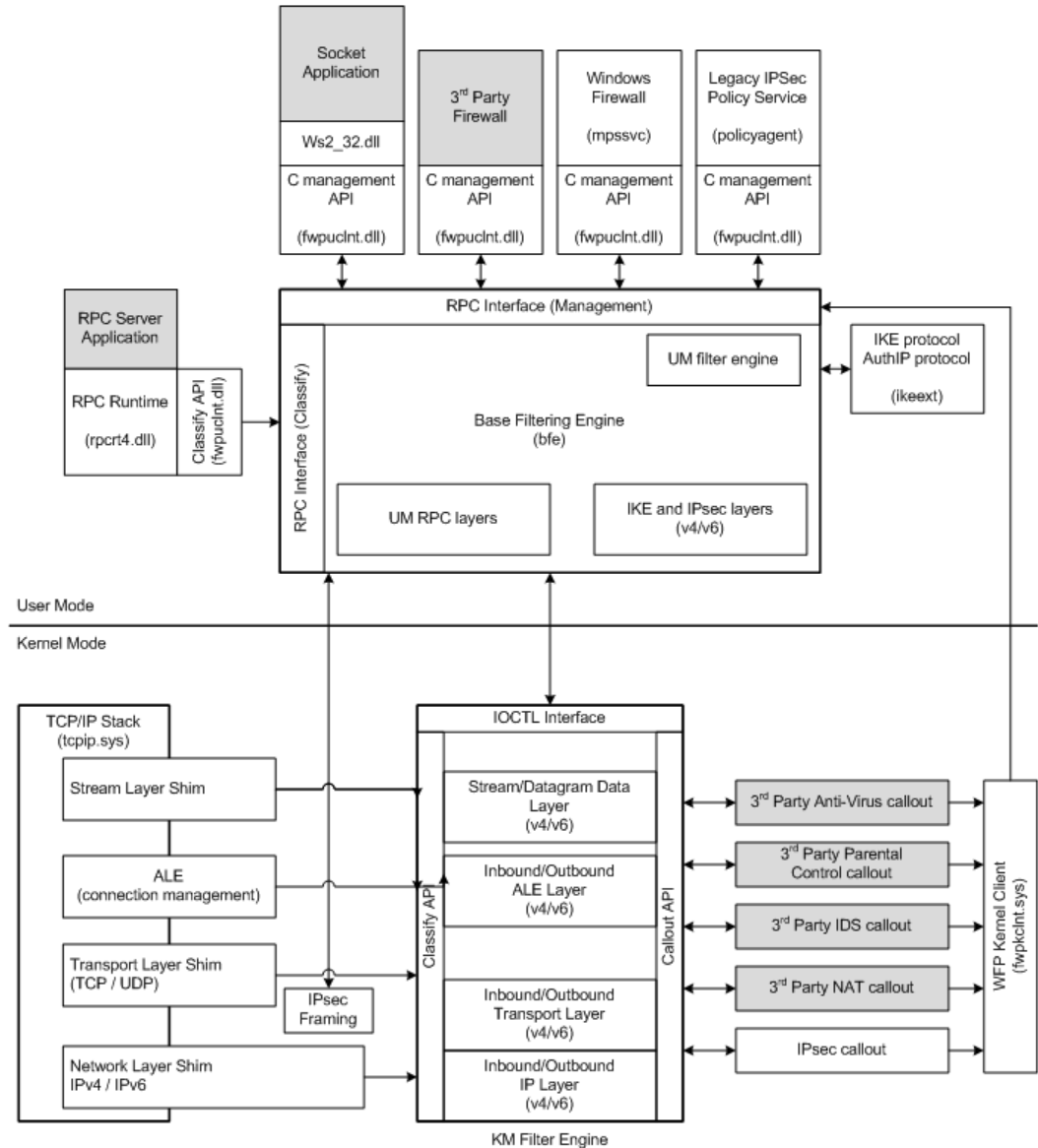
Source: <https://docs.microsoft.com/en-us/windows/desktop/fwp/windows-filtering-platform-architecture-overview>

# Windows Filtering Platform (WFP)

## ■ Callouts

- Callbacks that receive the entire packet for analysis or modification

- » Permit
- » Block
- » Continue
- » Defer
- » Need more data
- » Drop connection

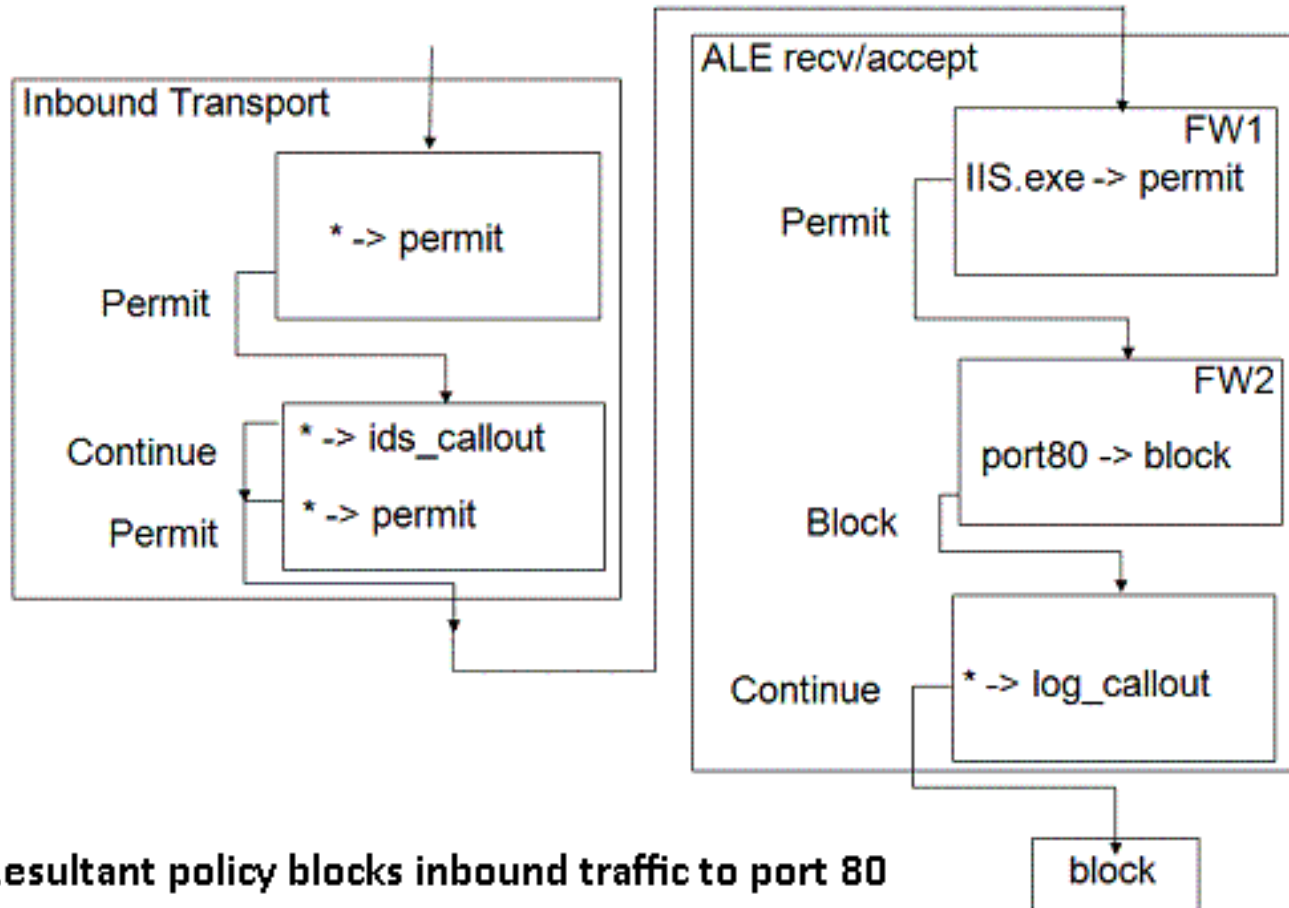


# Windows Filtering Platform (WFP)

---

- Filter arbitration (rules of evaluation)
  - Filters are evaluated in a pre-determined order
  - For each filter layer, sub-layers may be registered
    - » Sub-layers have a priority (weight) value (assigned by the developers)
      - Higher priority = earlier evaluation
  - Sub-layers consist of rules
    - » Rules are also weighted
    - » Block / Permit stops the evaluation within the current sub-layer
    - » The verdict can be flagged as a 'hard' result (hard block, hard permit)
  - Evaluation continues through all the sub-layers, even if a higher priority one determines that the input should be Blocked
    - » Soft permit < Soft block < Hard permit < Hard block ('Veto')
    - » Overriding a hard permit with a hard block generates an audit event
      - But if the hard block comes first, the hard permit is silently ignored
    - » The 'hardness' flag is cleared at the ends of layers

# Windows Filtering Platform (WFP)



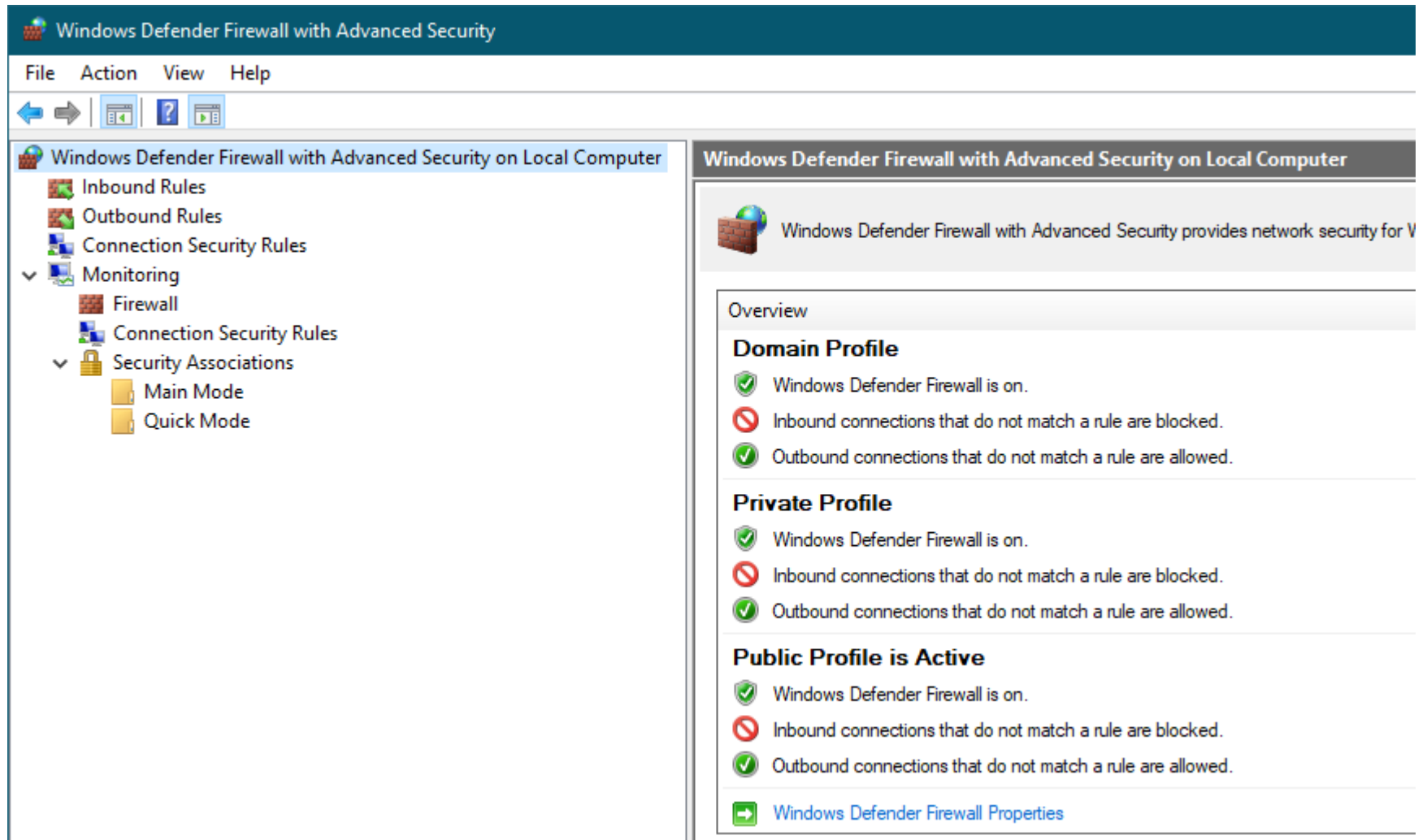
Source: <https://docs.microsoft.com/en-us/windows/desktop/fwp/filter-arbitration>

# Windows Firewall with Advanced Security

---

- The built-in firewall in Windows Vista and onwards
  - Renamed to Windows (Defender) Firewall with Advanced Security
- Split in two
  - Internet Connection Sharing (SharedAccess) – NAT functionality
  - Windows Firewall (MpsSvc) – Firewall functionality
    - » Renamed to Windows **Defender** Firewall in Windows 10, update 1709
- Relies on the Windows Filtering Platform
- Programmable
  - `netsh advfirewall`
  - PowerShell (Windows 8/Server 2012 and later versions)

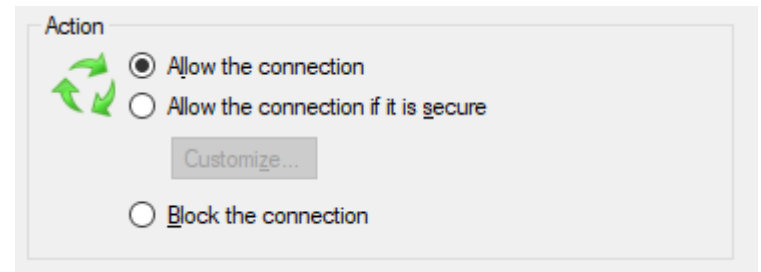
# Windows Firewall with Advanced Security



# Windows Firewall with Advanced Security

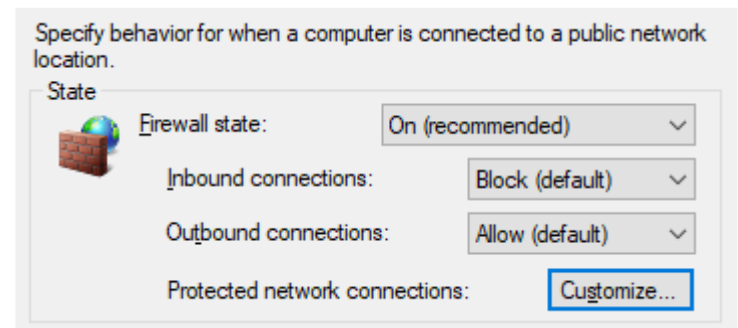
- Two (three) actions: Block and Allow (if secure)

- Block overrides Allow
- There is no 'Reject' (like in *iptables*)



- Direction-based filtering

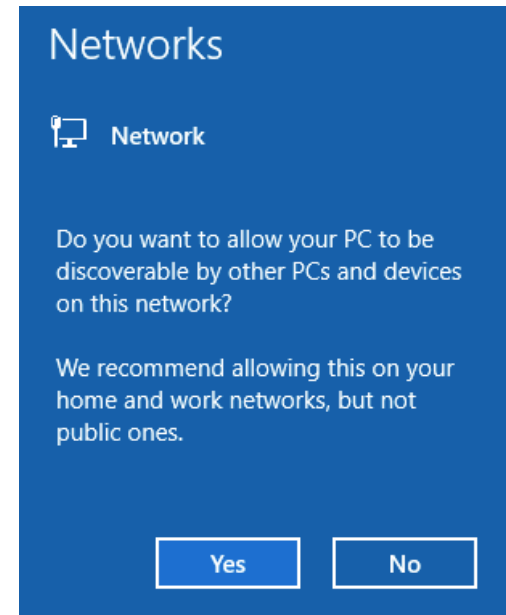
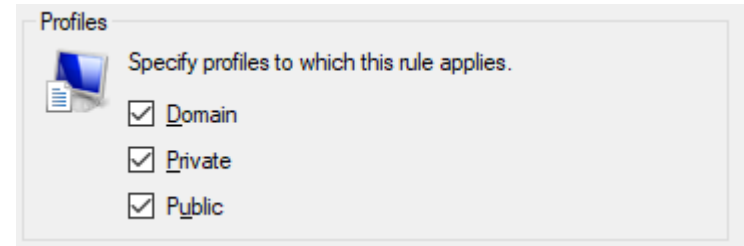
- Inbound (default: block)
- Outbound (default: allow)



# Windows Firewall with Advanced Security

## ■ Profile-based filtering

- Rules belong to 1..3 profiles
- Domain: for connections to and from the Active Directory domain network of which this computer is a member of
- Private: networks the user declared private
  - » Recommended for home or non-domain work networks
- Public: networks the user declared public
  - » Recommended for untrusted networks, such as hotel or airport WiFi networks

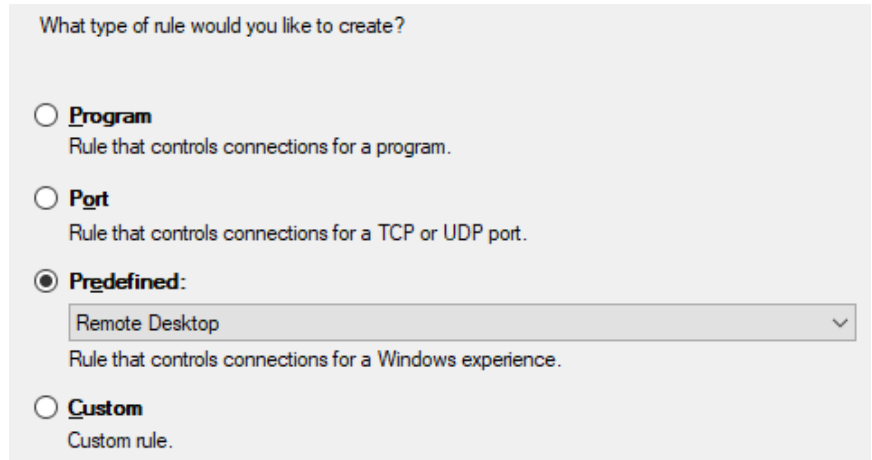




# Windows Firewall with Advanced Security

---

- Predefined rule groups
  - Make it easier to add rules for typical use-cases
  - Good for less tech-savvy users
- Rules can be enabled/disabled
  - Useful for debugging
- It is possible to Export/Import firewall configuration
  - Useful when migrating or for backup before a major reconfiguration



EXAMPLES

---

# **MANAGING WINDOWS FIREWALL**

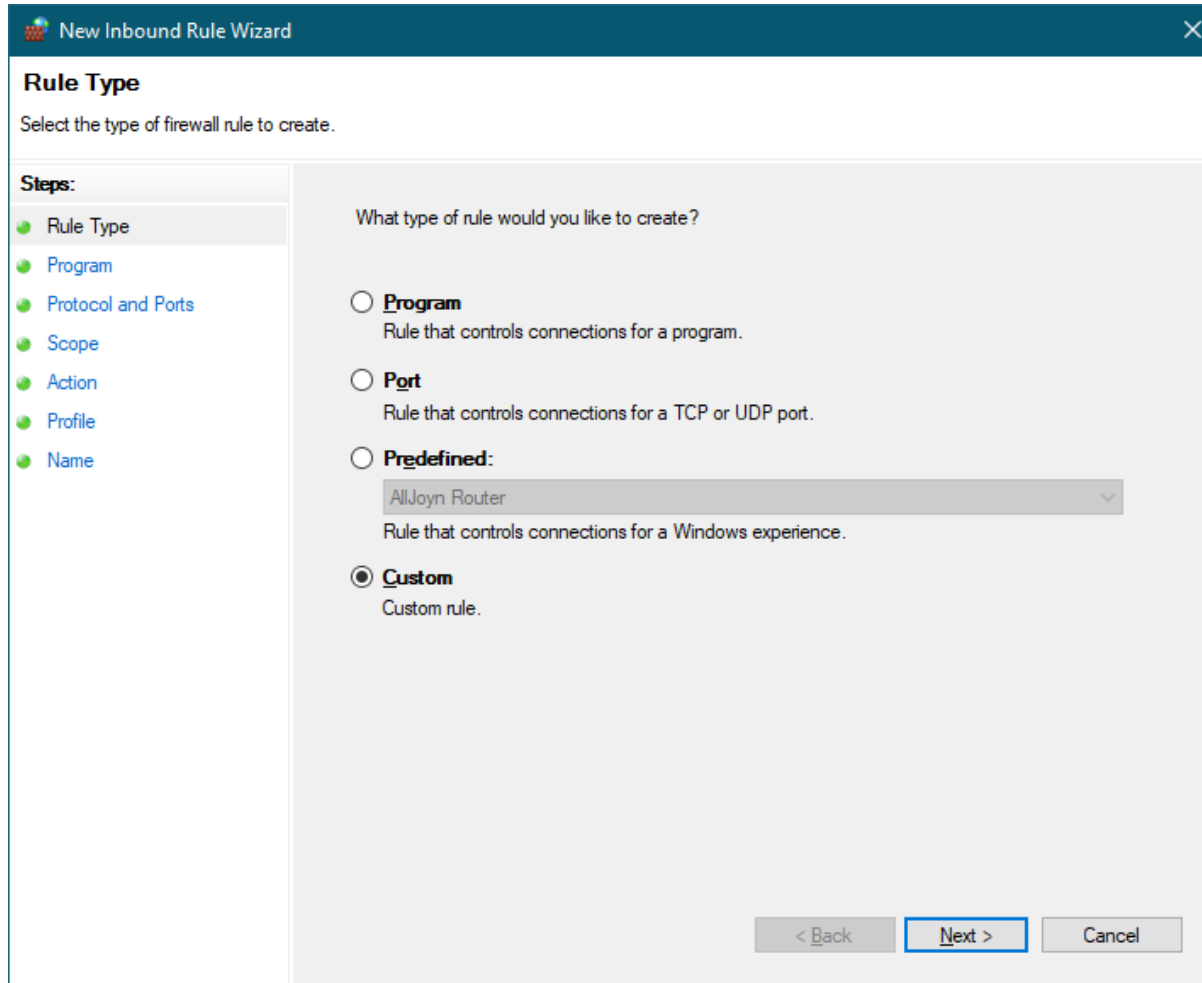
# Managing Windows Firewall

---

- The practical session begins here
- Further slides are only provided for reference,  
~~mainly for those who could not attend~~

# Management via the GUI

- Most easily launched via 'wf.msc'



# Management via the GUI

The screenshot shows the 'New Inbound Rule Wizard' window in Windows Firewall. The title bar is dark blue with a close button. The main area is white with a dark blue header. The 'Program' step is selected in the left sidebar. The main content area has a light gray background. The 'All programs' radio button is selected. The 'This program path' option is also visible with a text box and a 'Browse...' button. The 'Services' section is at the bottom with a 'Customize...' button. Navigation buttons '< Back', 'Next >', and 'Cancel' are at the bottom right.

**New Inbound Rule Wizard**

**Program**

Specify the full program path and executable name of the program that this rule matches.

**Steps:**

- Rule Type
- Program**
- Protocol and Ports
- Scope
- Action
- Profile
- Name

Does this rule apply to all programs or a specific program?

☒ **All programs**  
Rule applies to all connections on the computer that match other rule properties.

☐ **This program path:**

Example: c:\path\program.exe  
%ProgramFiles%\browser\browser.exe

**Services**

Specify which services this rule applies to.

# Management via the GUI

The screenshot shows the 'New Inbound Rule Wizard' window in Windows Firewall. The title bar reads 'New Inbound Rule Wizard'. The main heading is 'Protocol and Ports', with a subtitle 'Specify the protocols and ports to which this rule applies.' On the left, a 'Steps:' pane lists: Rule Type, Program, Protocol and Ports (highlighted), Scope, Action, Profile, and Name. The main area asks 'To which ports and protocols does this rule apply?'. It contains the following fields: 'Protocol type:' set to 'TCP'; 'Protocol number:' set to '6'; 'Local port:' set to 'Specific Ports' with a text box containing '80,443' and an example '80, 443, 5000-5010'; 'Remote port:' set to 'All Ports' with an example '80, 443, 5000-5010'. At the bottom, there is a section for 'Internet Control Message Protocol (ICMP) settings:' with a 'Customize...' button. Navigation buttons at the bottom right are '< Back', 'Next >' (highlighted), and 'Cancel'.

New Inbound Rule Wizard

**Protocol and Ports**

Specify the protocols and ports to which this rule applies.

**Steps:**

- Rule Type
- Program
- Protocol and Ports**
- Scope
- Action
- Profile
- Name

To which ports and protocols does this rule apply?

Protocol type: TCP

Protocol number: 6

Local port: Specific Ports

80,443

Example: 80, 443, 5000-5010

Remote port: All Ports

Example: 80, 443, 5000-5010

Internet Control Message Protocol (ICMP) settings: Customize...

< Back Next > Cancel

# Management via the GUI

The screenshot shows the 'New Inbound Rule Wizard' window in Windows Firewall. The title bar is dark blue with a close button. The main area is white. On the left, a 'Steps:' sidebar lists: Rule Type, Program, Protocol and Ports, Scope (selected), Action, Profile, and Name. The 'Scope' section is titled 'Scope' and contains the instruction 'Specify the local and remote IP addresses to which this rule applies.' The main content area has two sections. The first is 'Which local IP addresses does this rule apply to?' with two radio buttons: 'Any IP address' (unselected) and 'These IP addresses:' (selected). Below the selected option is a list box containing '192.168.50.100'. To the right of the list box are three buttons: 'Add...', 'Edit...', and 'Remove'. Below this is a text label 'Customize the interface types to which this rule applies:' followed by a 'Customize...' button. The second section is 'Which remote IP addresses does this rule apply to?' with two radio buttons: 'Any IP address' (selected) and 'These IP addresses:' (unselected). Below the selected option is an empty list box. To the right of the list box are three buttons: 'Add...', 'Edit...', and 'Remove'. At the bottom right are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

New Inbound Rule Wizard

**Scope**

Specify the local and remote IP addresses to which this rule applies.

**Steps:**

- Rule Type
- Program
- Protocol and Ports
- Scope**
- Action
- Profile
- Name

**Which local IP addresses does this rule apply to?**

☐ Any IP address

☒ These IP addresses:

192.168.50.100

Add... Edit... Remove

Customize the interface types to which this rule applies: Customize...

**Which remote IP addresses does this rule apply to?**

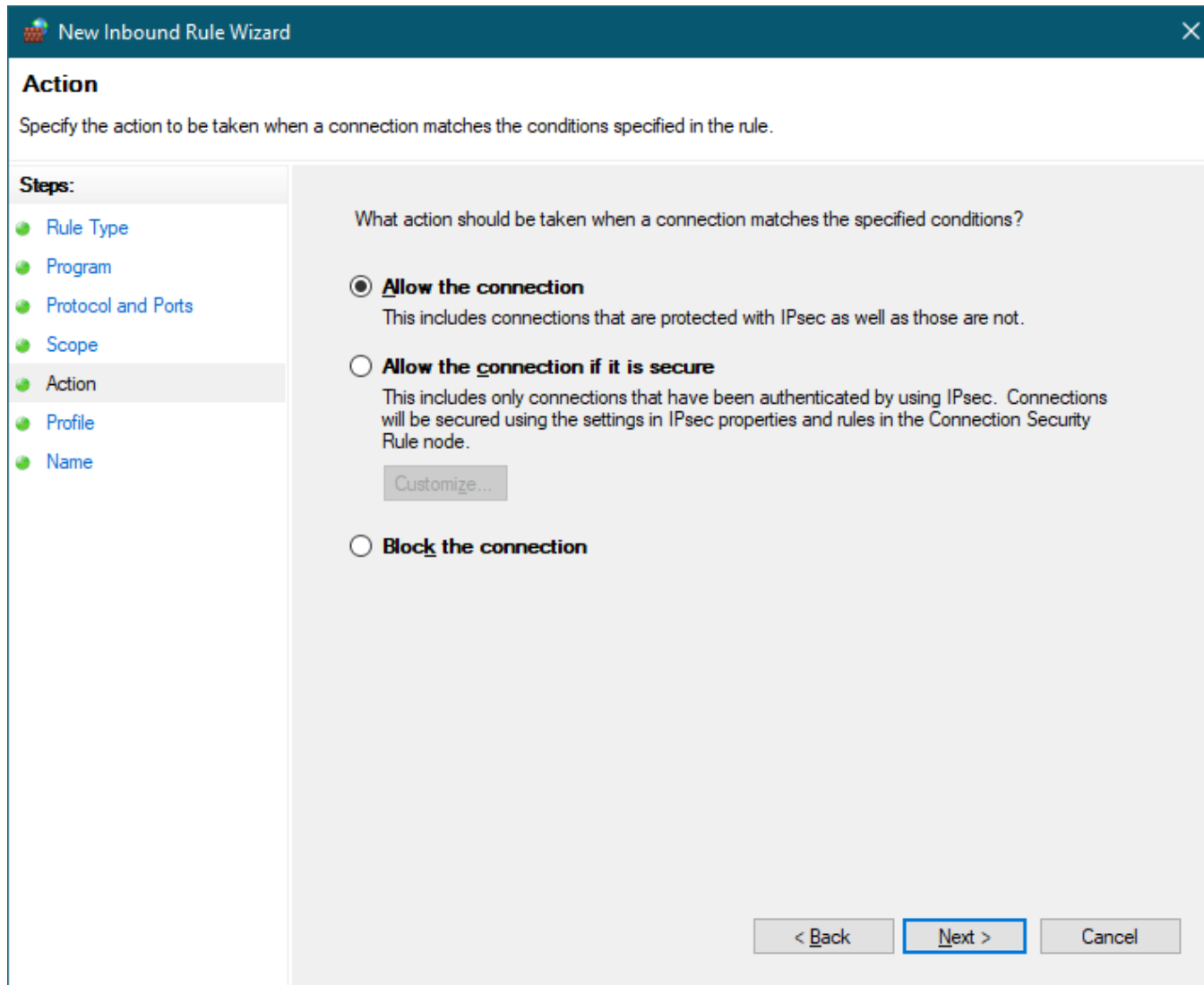
☒ Any IP address

☐ These IP addresses:

Add... Edit... Remove

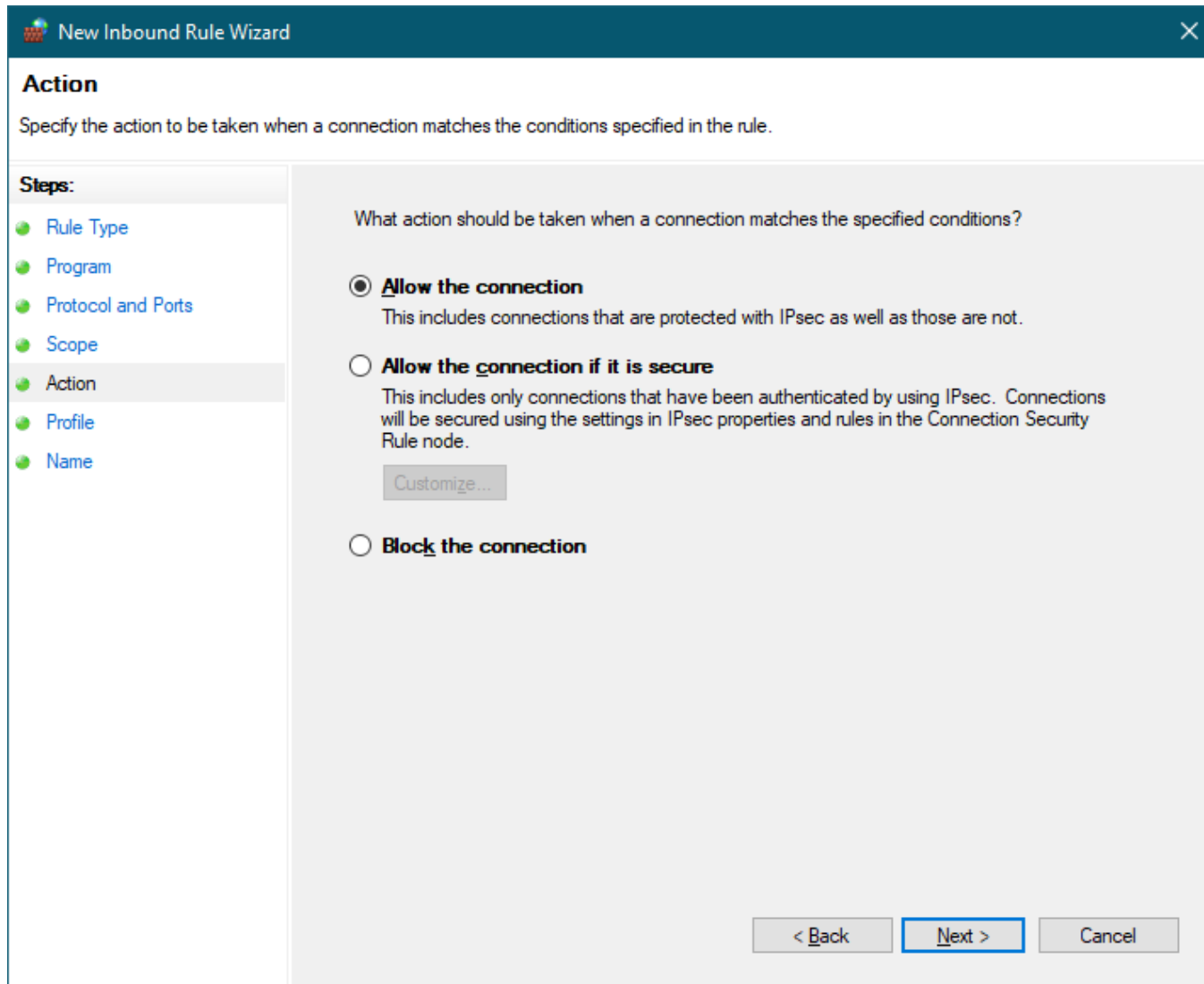
< Back Next > Cancel

# Management via the GUI

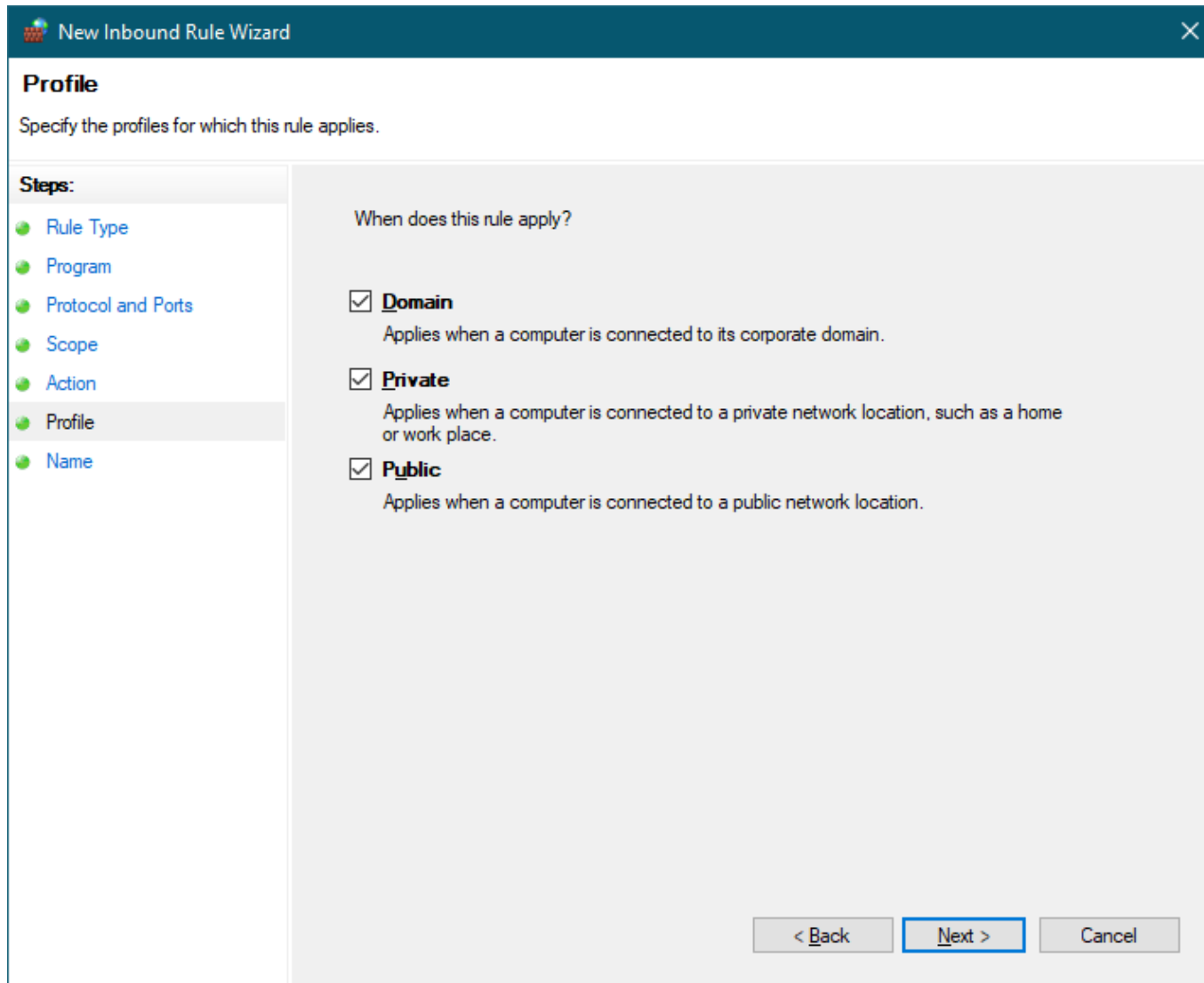




# Management via the GUI



# Management via the GUI



# Management via the GUI

**New Inbound Rule Wizard**

**Name**

Specify the name and description of this rule.

**Steps:**

- Rule Type
- Program
- Protocol and Ports
- Scope
- Action
- Profile
- Name**

**Name:**

Allow web traffic

**Description (optional):**

As requested in ticket #56123

< Back   **Finish**   Cancel

# Management via netsh

---

- `netsh advfirewall firewall add rule name="Allow web traffic in" localport=80,443 protocol=TCP dir=in action=allow`
- `netsh advfirewall firewall show rule name="Allow web traffic in"`  
Rule Name: Allow web traffic in  
-----  
Enabled: Yes  
Direction: In  
Profiles: Domain,Private,Public  
Grouping:  
LocalIP: Any  
RemoteIP: Any  
Protocol: TCP  
LocalPort: 80,443  
RemotePort: Any  
Edge traversal: No  
Action: Allow  
Ok.
- `netsh advfirewall firewall delete rule name="Allow web traffic in"`

# Management via Powershell

---

- `New-NetFirewallRule -DisplayName "Allow web traffic in" -Direction Inbound -Enabled True -Action Allow -Protocol TCP -LocalPort 80,443`
- `Get-NetFirewallRule -DisplayName "Allow web traffic in"`  
Name : {c4a4b5f9-41f3-47e8-8959-8d714e04e3ef}  
DisplayName : Allow web traffic in  
Description :  
Group :  
Enabled : True  
Profile : Any  
Direction : Inbound  
Action : Allow  
EdgeTraversalPolicy : Block  
PrimaryStatus : OK  
Status : The rule was parsed successfully from the store. (65536)  
PolicyStoreSource : PersistentStore  
PolicyStoreSourceType : Local
- `Get-NetFirewallRule -DisplayName "Allow web traffic in" | Remove-NetFirewallRule`

TASKS

---

# **MANAGING WINDOWS FIREWALL**

# Tasks

---

- Set up the virtual machines
  - Create two VMs from the .ova provided
  - Name one of them 'client', the other, 'server'
  - Make sure they're assigned to the same internal network
  - Start the machines
  
- Set up the client and the server
  - Server
    - » IP address: 10.10.100.200/24
    - » Start the DNS server service: `sc start DNS`
  - Client
    - » IP address: 10.10.100.1/24
    - » DNS server: 10.10.100.200

# Tasks

---

- Try pinging the server from the client (ping 10.10.100.200 -t)
  - Why isn't it working?
- Check the firewall rules on the server
  - Familiarize yourself with the GUI
  - Enable File and Printer Sharing (Echo Request – ICMPv4-In)
  - What profile is this enabled for?
- Delete all incoming and outgoing firewall rules
- Disable the firewall on the client



# Tasks

---

- Allow some incoming ICMP traffic
  - Only from within the subnet 10.10.100.0/24
  - Only the following ICMP types
    - » Packet Too Big
    - » Destination Unreachable
    - » Source Quench
    - » Echo Request
    - » Time Exceeded
  
- Can you ping the server now?
  
- Start ping the client from the server (use -t)

# Tasks

---

- Set the firewall to block all outgoing traffic (unless allowed)
  - Can you still ping the server?
  - Can you still ping the client?
- Set the firewall to explicitly block all outgoing ICMP traffic
  - Can you still ping the server?
  - Can you still ping the client?
- Key take-away: outgoing packets belonging to an allowed connection are NOT affected by blocking rules
  - How does this compare to allowing such traffic using iptables?

# Tasks

---

- Start pingging the server from itself
  - In one window, use 127.x.y.z
  - In another, use 10.10.100.200
- What's happening?
- Key take-away: traffic to and from loopback addresses and local IP addresses is not filtered by Windows Firewall
  - How does this compare to allowing such traffic using iptables?

# Tasks

---

- Start Lighttpd on the server
- Load <http://www.mysite.local> on the client
  - It's not working... what could be the problem?
  - Load it on the server... it works!
- Maybe something needs to be permitted through the firewall?
  - Turn on logging to see what's happening
    - » Apparently, UDP 53 is important for some reason...
  - Allow UDP 53
- How about now?
  - Of course, we need TCP 80...
  - Allow TCP 80
- Yey!

# Tasks

---

- Stop Lighttpd
- Start IIS
  - Can you see IIS's welcome page?
- Modify the rule that permits TCP 80 to permit only Lighttpd to accept connections
  - Can you still see IIS's welcome page?
- Stop IIS
- Start Lighttpd
  - Can you see Lighttpd's welcome page?

# Tasks

---

- Attempt to access the C\$ admin share on [\\10.10.100.200](#)
  - Are you successful?
- Allow SMB access using the premade rules wizard
  - Are you successful now?

# Tasks

---

- On the server, delete the rule that allows incoming DNS traffic
- On the client, try resolving [www.mysite.local](http://www.mysite.local) using *nslookup*
- Using netsh, add a rule that allows incoming DNS traffic
  - `netsh advfirewall firewall add rule name="Allow DNS in" localport=53 protocol=udp dir=in action=allow`
- Was it added successfully?
  - Check it using netsh
    - » `netsh advfirewall firewall show rule dir=in name=all`
  - Check it using the GUI
  - Try nslookup again
- Delete the rule using netsh
  - `netsh advfirewall firewall delete rule name="Allow DNS in"`
  - Try nslookup again

# Tasks

---

- Again, add a rule that allows inbound DNS traffic
  - This time, use Powershell
  - `New-NetFirewallRule -DisplayName "Allow DNS in" -Protocol UDP -LocalPort 53 -Direction Inbound`
- Verify that it was added
  - `Get-NetFirewallRule -DisplayName "Allow DNS in"`
  - Use the GUI
  - Use nslookup
- Delete the rule using Powershell
  - `Get-NetFirewallRule -DisplayName "Allow DNS in" | Remove-NetFirewallRule`



# Tasks

---

- Export the current firewall configuration to *backup.wfw*
- Delete all inbound rules
  - Oops...
- Restore configuration using the backup
- Optional, if time permits
  - Add a rule using the Local Group Policy Editor and see how it appears, whether you can edit/delete it
  - Observe what you can see in Event Viewer for the Windows Firewall

---

**MISCELLANEOUS**

# Further Reading

---

- Kamel Messaoudi: Network traffic filtering technologies for Windows  
<https://briolidz.wordpress.com/2011/12/20/network-traffic-filtering-technologies-for-windows/>
- NDIS-hooking drivers and legacy Windows systems  
<http://omegadroid.co/wanted-knox-void-warranty-0x1/>
- About Windows Filtering Platform  
<https://docs.microsoft.com/en-us/windows/desktop/fwp/about-windows-filtering-platform>
- Filter Arbitration  
<https://docs.microsoft.com/en-us/windows/desktop/fwp/filter-arbitration>

## Inbound Rules

- [illegible]

# Control Questions

---

- Has Windows always had a built-in firewall?
- Is Windows' firewall a static or a stateful firewall? Why?
- Briefly explain what the Windows Filtering Platform is and how it works. (It is recommended that you draw a sketch.)
- What is the purpose of the Base Filtering Engine?
- What is the purpose of the shims (in the context of the WFP)?
- What is the purpose of the callouts (in the context of the WFP)?

# Control Questions

---

- Name two similarities and two differences between iptables and the modern Windows Firewall.
- What actions (decisions) are there in the modern Windows Firewall for packet handling?
- What profiles are there in the modern Windows Firewall? What is the purpose of each?
- By default, how are incoming and outgoing packets handled by a modern Windows Firewall?

# Control Questions

---

- How does the modern Windows Firewall handle packets that are responses to permitted incoming packets?
- How does the modern Windows Firewall handle packets that are addressed to and from loopback addresses?
- How can the modern Windows Firewall be managed?
- Briefly explain how you would add a new rule to a modern Windows Firewall, using the graphical user interface.