Контрол на достъпа до услугите

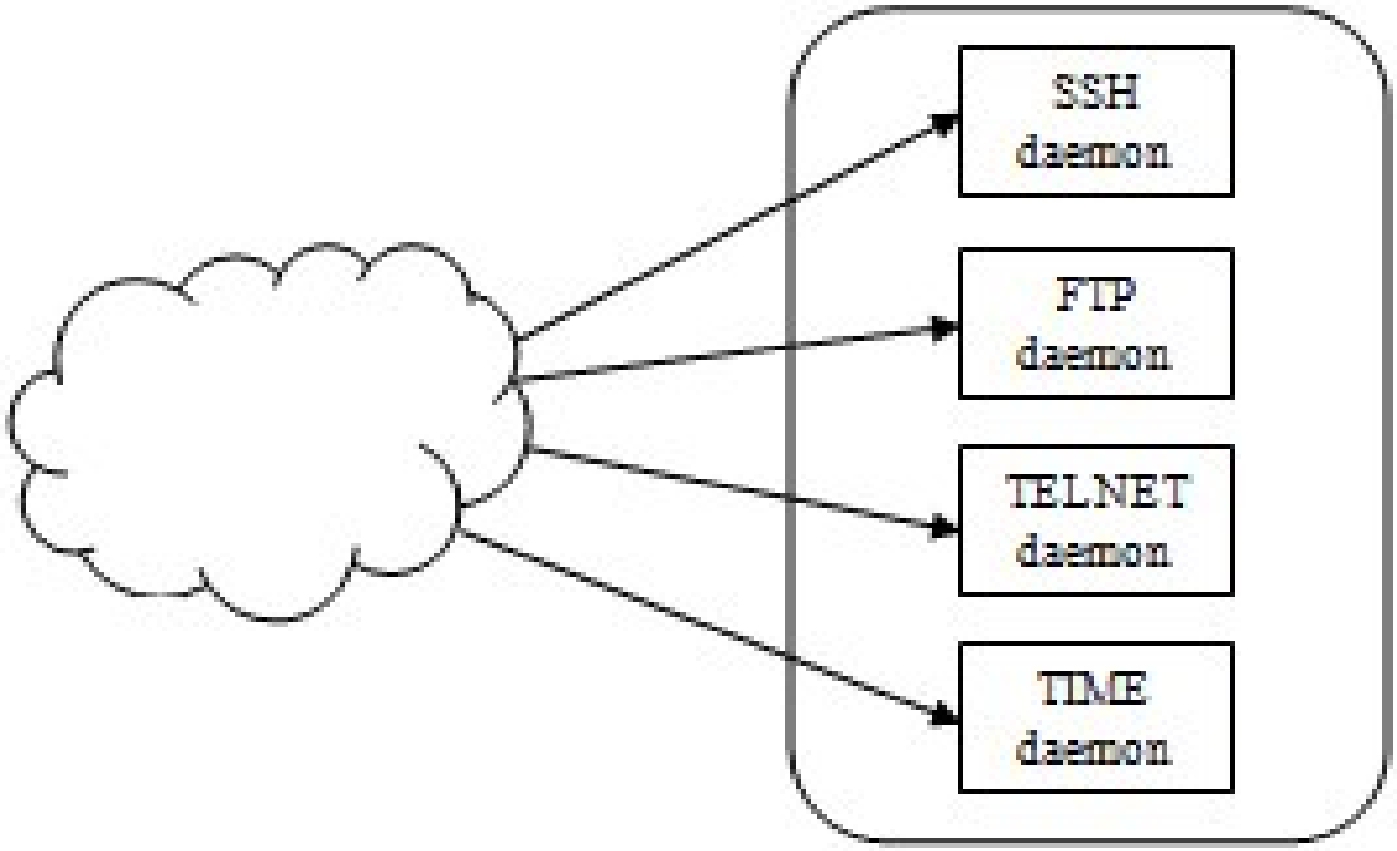
Контрол на достъпа до услугите

Контролът на достъпа е техника за ограничаване на достъпа до съответни услуги.

Контрол на достъпа до услугите

- Контролът на достъпа е техника за ограничаване на достъпа до съответни услуги.
- Ограничаването може да бъде по:
  - IP адрес на хост/мрежа
  - Име на хост, изискващ дадена услуга
  - Име на домейн
  - и др.
- Осъществява се на базата на контролен списък за достъп:
  - Ако в списъка е позволено на хоста да използва услугата, заявката се разрешава.
  - В противен случай тя се отхвърля.

Система със самостоятелно стартирани услуги



Система със самостоятелно стартирани услуги

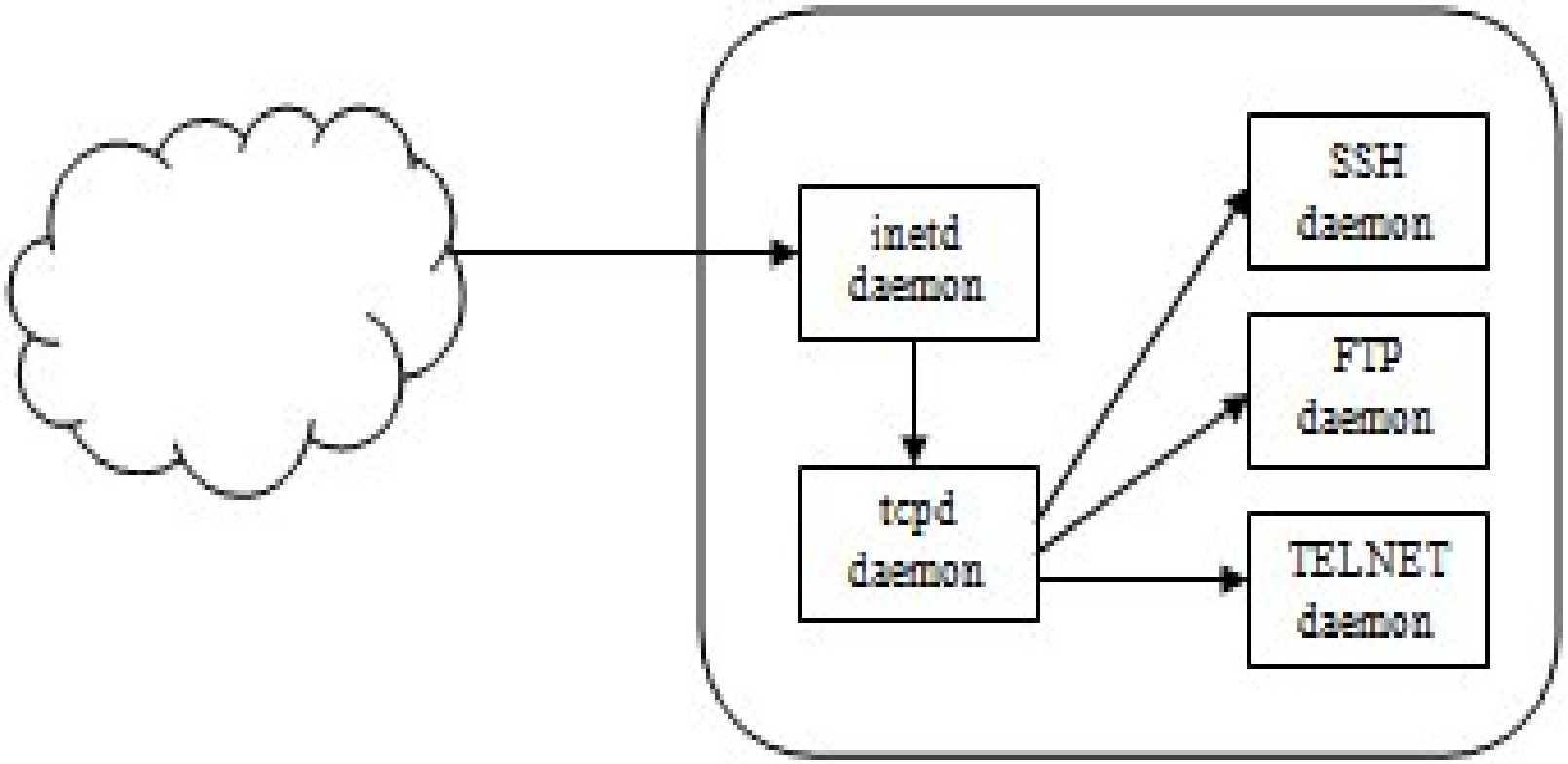
- Стартирани са десетки или стотици демони
- Всяка услуга се обслужва от свой демон:
  - след като обслужи процеса, заспива
  - в по-голямата част от времето очакват (а не изпълняват) заявки,

  Недостатък:
  - заемат ресурс памет

  Предимство:
  - сървърите са оптимизирани за тази услуга

➢ Не е ефективно – излишно разходване на памет

Система с централна услуга



Система с централна услуга

- Стартиран е един централен демон
- Под Linux това е супер-сървър – ***inetd***.
- Той очаква заявки по всички портове и стартира за тяхното обслужване само необходимите услуги:
  - Постъпва заявка
  - Създава дъщерен процес със съответния за услугата сървър
  - Предава му за обработка тази конекция
  - След като обработи напълно заявката, завършва.

➢ По-ефективно – само 1 сървър е в стадий на обработка и не се дублира памет

inetd супер-сървър

/etc/inetd.conf

service type protocol wait user server cmdline

Конфигуриране на inetd

- ***Service*** – името на услугата (преобразува се в номер на порт, съгласно */etc/services)*
- ***Type*** – типа на използвания сокет (*stream* за TCP и *dgram* за UDP)
- ***Protocol*** – името на транспортния протокол (да е дефинирано в */etc/protocols*)
- ***Wait*** :
  - *wait* - ***inetd*** стартира само един сървър за съответния порт, като изчаква неговото завършване за повторно обслужване на заявка
  - *nowait* - ***inetd*** незабавно продължава да очаква заявки по същия порт след стартирането на услугата.
- ***User*** – името на потребителя, който ще бъде собственик на създадения дъщерен процес
- ***Server*** – пълния път до изпълнимата сървърна програма
- ***Cmdline*** – командният ред, който ще бъде предаден като параметър на стартирания сървър (името на сървъра и аргументите за предаване)

inetd супер-сървър

```
# discard       dgram   udp     wait    root    internal
# daytime       stream  tcp     nowait  root    internal
# daytime       dgram   udp     wait    root    internal
# chargen       stream  tcp     nowait  root    internal
# chargen       dgram   udp     wait    root    internal
#time   stream  tcp     nowait  root    internal
#time   dgram   udp     wait    root    internal
#
# These are standard services.
#
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  wu.ftpd -l -i -a
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  proftpd
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
#
# telnet stream  tcp     nowait  root    /usr/sbin/tcpd  /usr/sbin/in.telnetd
#
```

/etc/services

```
chargen         19/udp      ttytst      source    #Character Generator
ftp-data        20/tcp      #File Transfer [Default Data]
ftp-data        20/udp      #File Transfer [Default Data]
ftp             21/tcp      #File Transfer [Control]
ftp             21/udp      #File Transfer [Control]
ssh             22/tcp      #Secure Shell Login
ssh             22/udp      #Secure Shell Login
telnet          23/tcp
telnet          23/udp
#               24/tcp      any private mail system
#               24/udp      any private mail system
smtp            25/tcp      mail        #Simple Mail Transfer
smtp            25/udp      mail        #Simple Mail Transfer
```

/etc/protocols

```
ip              0       IP      # internet protocol, pseudo protocol
number
icmp            1       ICMP    # internet control message protocol
igmp            2       IGMP    # internet group management protocol
ggp             3       GGP     # gateway-gateway protocol
ipencap         4       IP-ENCAP        # IP encapsulated in IP
st2             5       ST2     # ST2 datagram mode (RFC 1819)
tcp             6       TCP     # transmission control protocol
```

TCP wrapper

- Контрол на достъпа на ниво хост.
- Използват се два файла:

/etc/hosts.allow

/etc/hosts.deny

servicelist : host-list [: shellcmd]

TCP wrapper

```
# /etc/hosts.deny
ALL: ALL
```

- # /etc/hosts.allow
- proftpd: .tu-varna.bg
- in.telnetd: 194.141.24.15, 194.141.25.

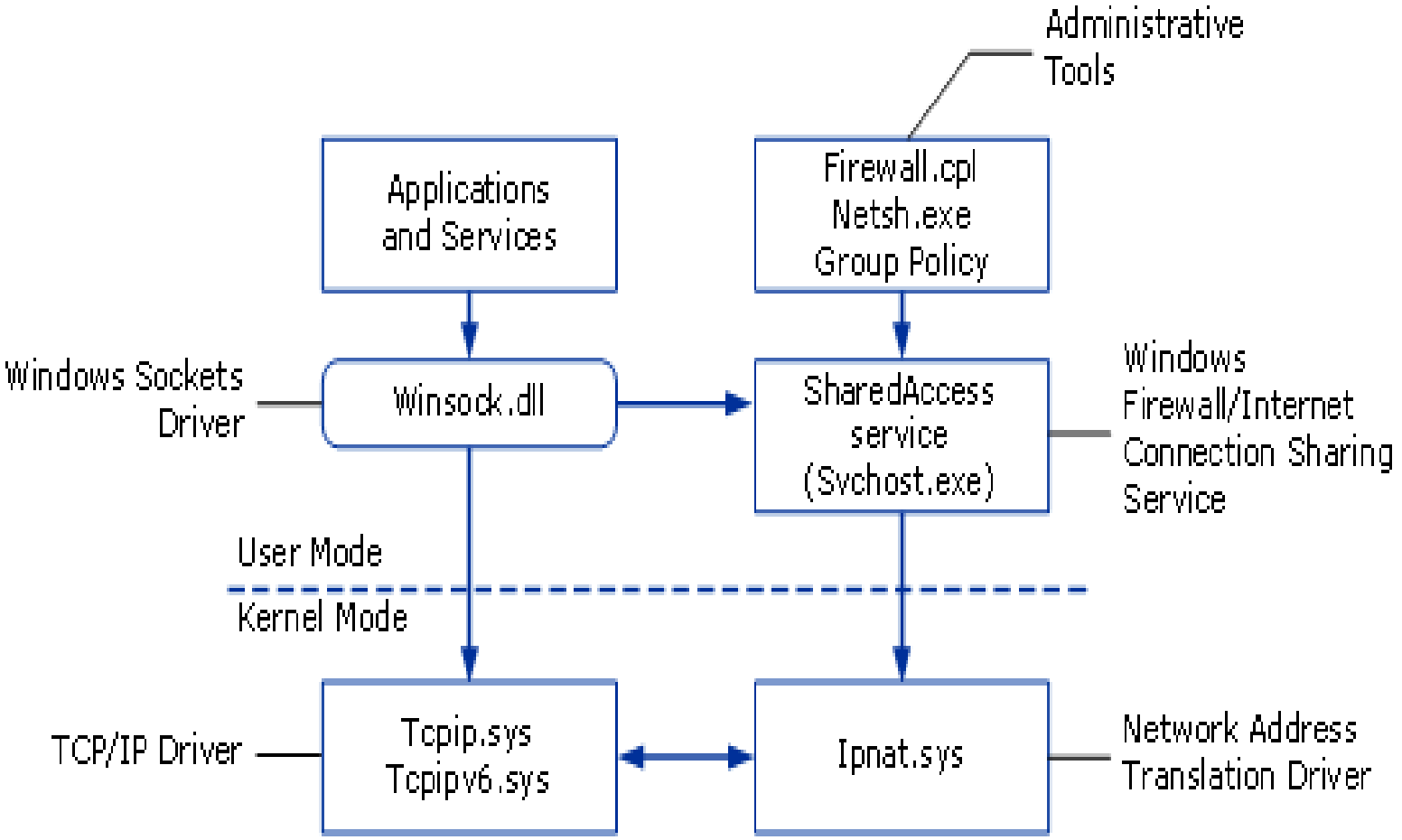
Контрол на достъпа в Windows

- **Windows Defender Firewall** – stateful технология, инспектираща и филтрираща IP трафика.
- Запомня състоянието на всяка конекция.
- Основно блокира идващи конекции.
- Използва изключения за разрешаване на идващи конекции.

Контрол на достъпа в Windows

- **Windows Defender Firewall** – stateful технология, инспектираща и филтрираща IP трафика.
- Запомня състоянието на всяка конекция.
- Основно блокира идващи конекции.
- Използва изключения за разрешаване на идващи конекции.

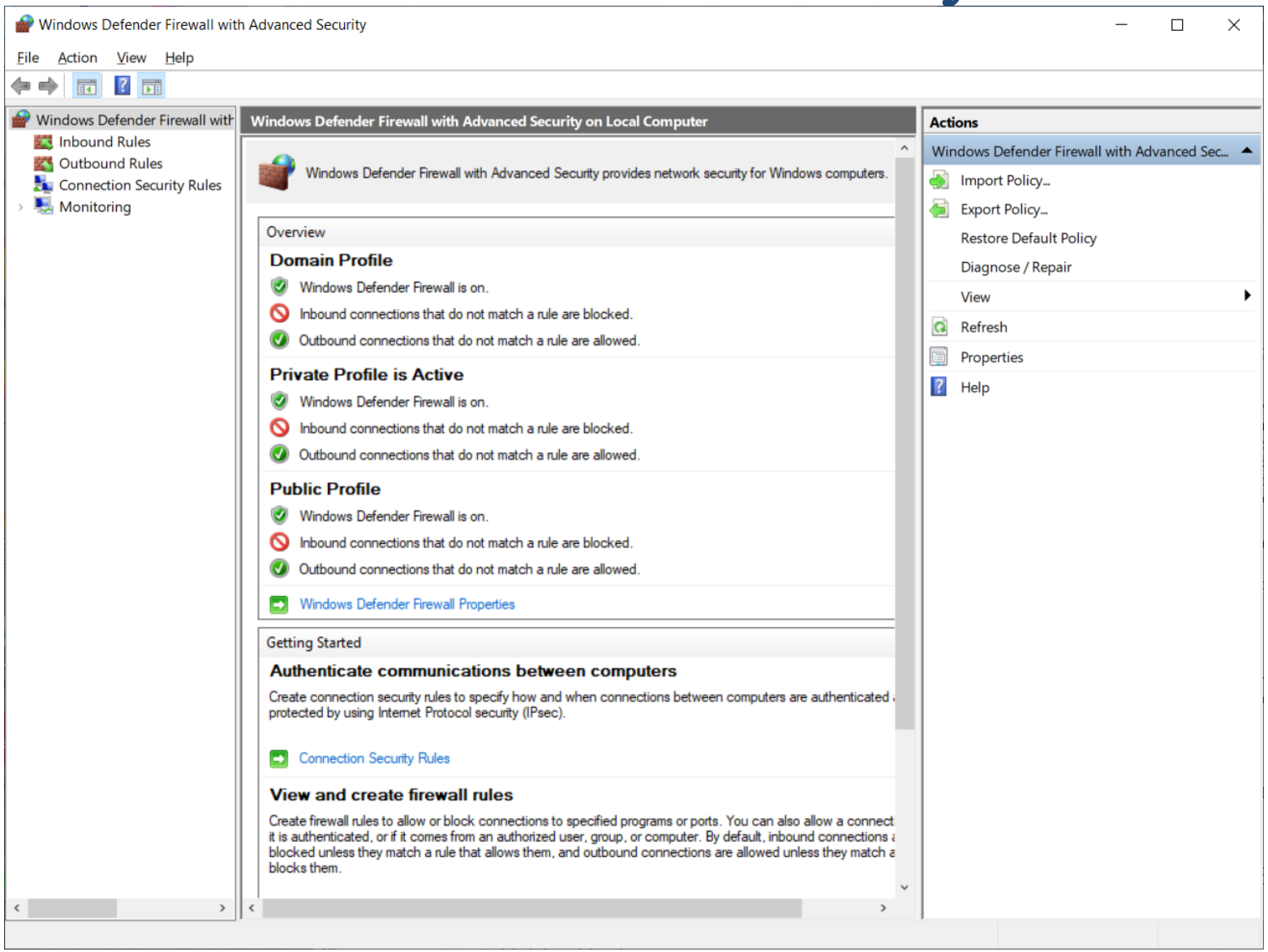
Windows Firewall



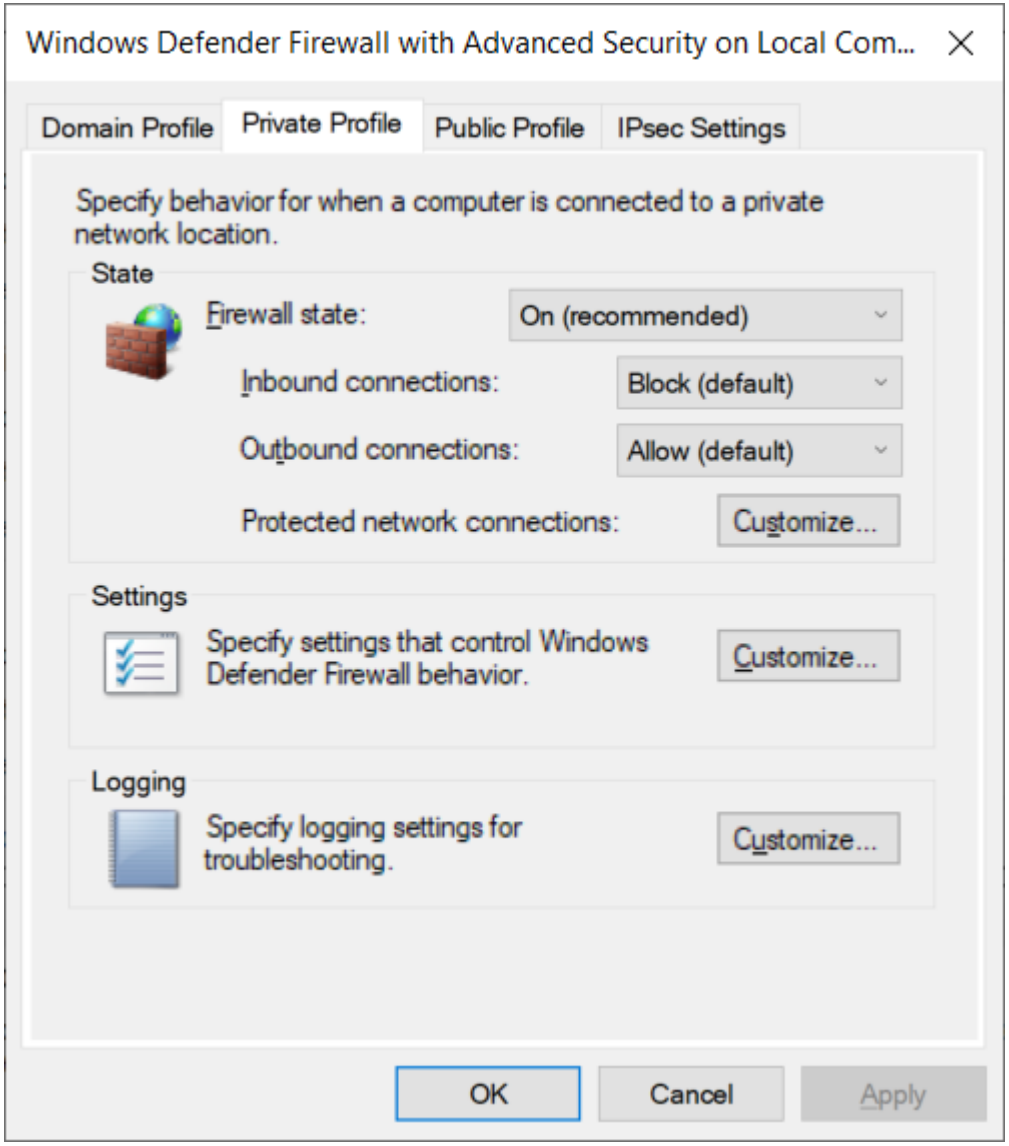
Windows Defender Firewall with Advanced Security

- Domain profile – настройки, прилагани когато машината е свързана в мрежа с домейн контролер.
- Private profile – когато мрежата не е директно свързана с Интернет.
- Public profile – при връзка с публична мрежа.

Windows Defender Firewall with Advanced Security

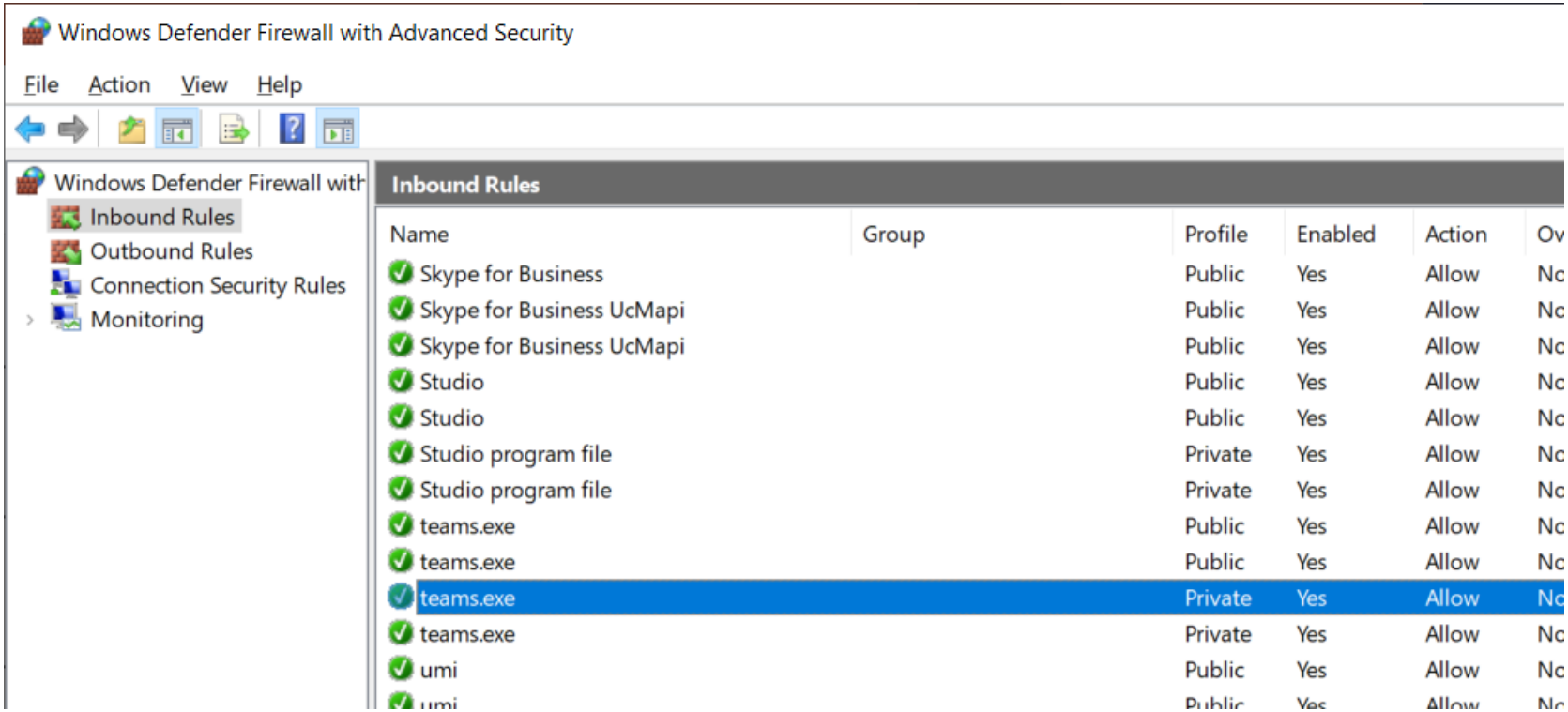


Модификация на профили

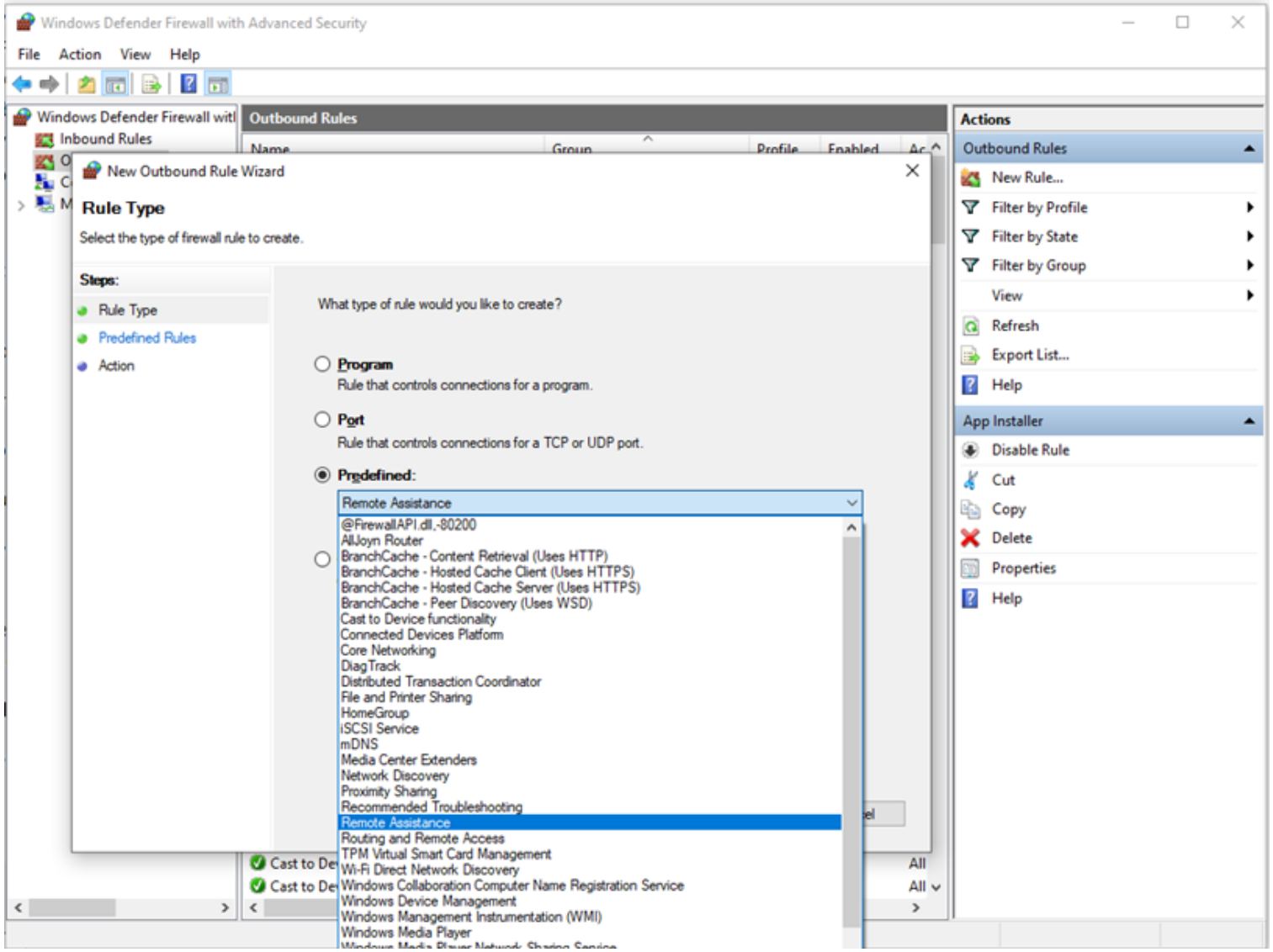


Контролиране на трафика

- Правила за идващи (inbound) конекции
- Правила за излизащи (outbound) конекции



Създаване на правила



Принципи

- Явно дефинираните правила имат предимство пред подразбиращите с настройки за блокиране.
- Явните блокиращи правила имат предимство пред някои конфликтни разрешаващи правила.
- Повече специфичните правила имат предимство пред по-малко специфичните.

Въпроси ?