

# Проверяем доступность порта UDP с помощью утилиты PortQry

windowsnotes.ru/programs/proveryaem-dostupnost-porta-udp-s-pomoshhyu-utility-portqry

2 декабря 2024 г.

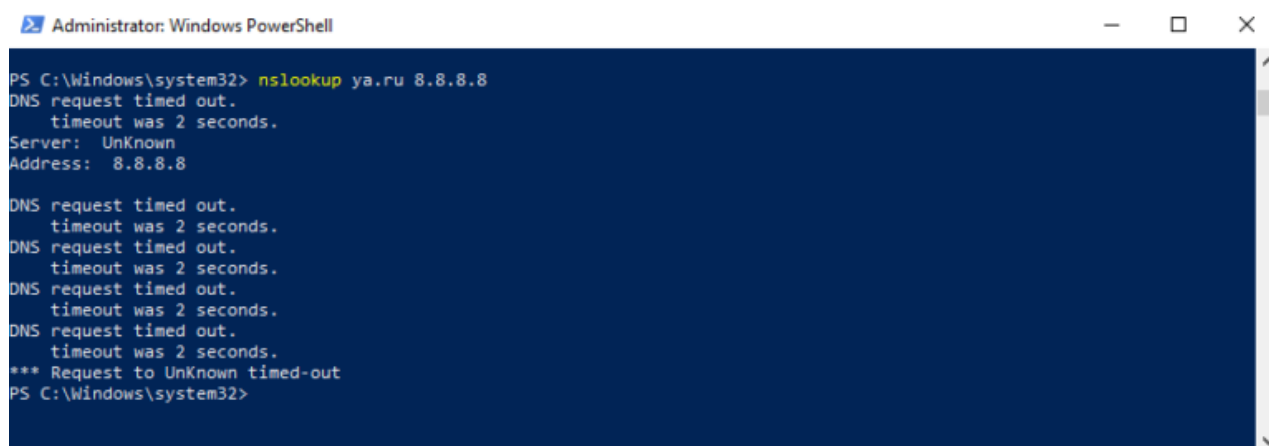
02.12.2024

Рубрики: [Программы](#)

В операционных системах Windows есть множество инструментов для диагностики сетевых проблем, но иногда этих инструментов бывает недостаточно. Поэтому сегодня рассмотрим утилиту PortQry, предназначенную для устранения неполадок с сетевыми подключениями. Рассматривать будем на конкретном примере, взятом из практики.

Итак, есть подозрения на проблему с DNS, и нам надо ее решить. Симптомы — не разрешаются внешние DNS-имена. В качестве основного DNS указан сервер Google (8.8.8.8) и нам надо проверить его доступность.

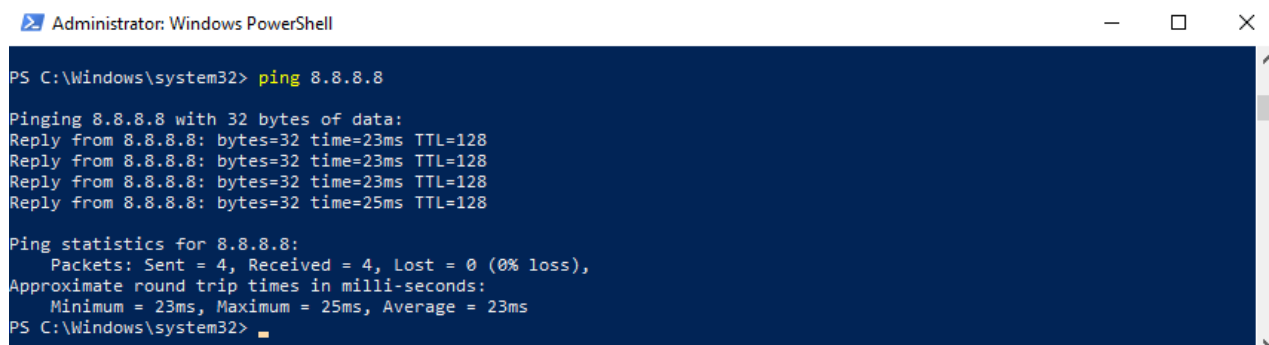
Для диагностики работоспособности DNS-сервера отправляем на него запрос с помощью утилиты nslookup и видим, что он не отвечает, соответственно адреса внешних ресурсов не резолвятся.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> nslookup ya.ru 8.8.8.8
DNS request timed out.
    timeout was 2 seconds.
Server: UnKnown
Address: 8.8.8.8

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out
PS C:\Windows\system32>
```

Проверяем доступность сервера. Пинг проходим успешно.

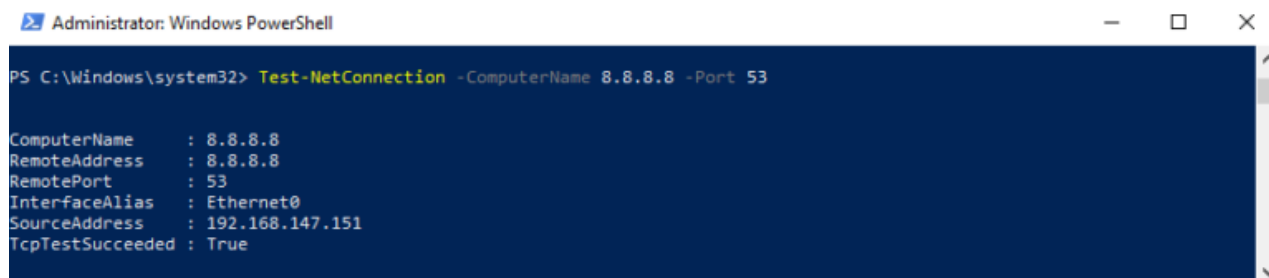


```
Administrator: Windows PowerShell
PS C:\Windows\system32> ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=23ms TTL=128
Reply from 8.8.8.8: bytes=32 time=23ms TTL=128
Reply from 8.8.8.8: bytes=32 time=23ms TTL=128
Reply from 8.8.8.8: bytes=32 time=25ms TTL=128

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 23ms, Maximum = 25ms, Average = 23ms
PS C:\Windows\system32>
```

Для проверки портов я обычно использую telnet, но сейчас для наглядности возьму PowerShell командлет Test-NetConnection. Проверяем доступность сервера по 53 порту и видим, что он открыт.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Test-NetConnection -ComputerName 8.8.8.8 -Port 53

ComputerName      : 8.8.8.8
RemoteAddress     : 8.8.8.8
RemotePort        : 53
InterfaceAlias    : Ethernet0
SourceAddress     : 192.168.147.151
TcpTestSucceeded  : True
```

Пока все идет по плану 😊

Но Test-NetConnection, как и горячо любимый мной telnet-клиент, умеет проверять только протокол TCP, а DNS-сервера для запросов используют UDP.

**Примечание.** Поскольку UDP-пакеты имеют небольшой размер, и не могут превышать 512 байт, для передачи данных, превышающих 512 байт, требуется протокол TCP. Поэтому DNS использует TCP для передачи зон, а для обычного разрешения имен используется UDP.

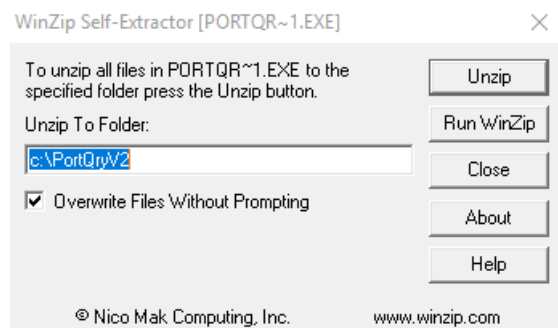
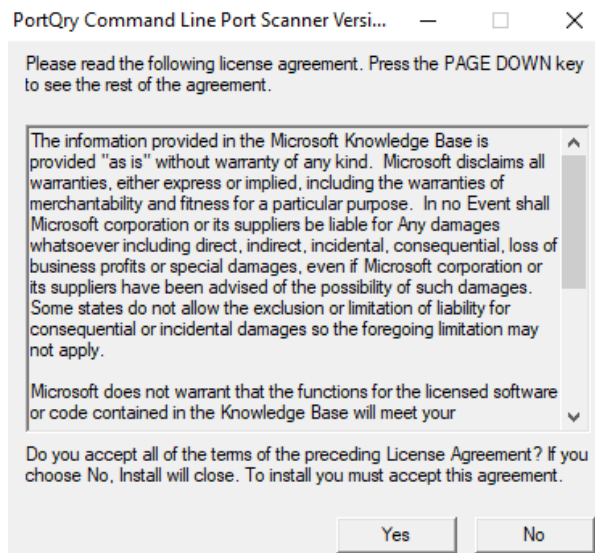
И вот тут на сцену выходит PortQry. Это утилита командной строки, предназначенная для диагностики проблем с сетевой доступностью. Она умеет проверять состояние портов TCP и UDP как на локальном, так и на удаленном компьютере.

Сторонней утилиту назвать не поворачивается язык, поскольку ее разработчиком является сама компания Microsoft. Соответственно скачать утилиту можно с их официального сайта. На данный момент доступна вторая версия утилиты PortQryV2.

Установки утилита не требует, а дистрибутив представляет из себя обычный самораспаковывающийся архив. Для использования PortQry надо согласиться с лицензионным соглашением

и распаковать утилиту в любое удобное место.

Дальше остается только перейти в каталог с утилитой и запустить ее. Запуск без параметров выводит справочную информацию.



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

PS C:\Windows\system32> cd C:\PortQryV2\
PS C:\PortQryV2> .\PortQry.exe

PortQry version 2.0

Displays the state of TCP and UDP ports

Command line mode: portqry -n name_to_query [-options]
Interactive mode:  portqry -i [-n name_to_query] [-options]
Local Mode:       portqry -local | -wpid pid | -wport port [-options]

Command line mode:

portqry -n name_to_query [-p protocol] [-e || -r || -o endpoint(s)] [-q]
        [-l logfile] [-sp source_port] [-sl] [-cn SNMP community name]

Command line mode options explained:
-n [name_to_query] IP address or name of system to query
-p [protocol] TCP or UDP or BOTH (default is TCP)
-e [endpoint] single port to query (valid range: 1-65535)
-r [end point range] range of ports to query (start:end)
-o [end point order] range of ports to query in an order (x,y,z)
-l [logfile] name of text log file to create
-y overwrites existing text log file without prompting
-sp [source port] initial source port to use for query
-sl 'slow link delay' waits longer for UDP replies from remote systems
-nr by-passes default IP address-to-name resolution
    ignored unless an IP address is specified after -n
-cn specifies SNMP community name for query
    ignored unless querying an SNMP port
    must be delimited with !
-q 'quiet' operation runs with no output
    returns 0 if port is listening
    returns 1 if port is not listening
    returns 2 if port is listening or filtered

Notes: PortQry runs on Windows 2000 and later systems
Defaults: TCP, port 80, no log file, slow link delay off
Hit Ctrl-c to terminate prematurely

examples:
portqry -n myserver.com -e 25
portqry -n 10.0.0.1 -e 53 -p UDP -i
portqry -n host1.dev.reskit.com -r 21:445
portqry -n 10.0.0.1 -o 25,445,1024 -p both -sp 53
portqry -n host2 -cn !my community name! -e 161 -p udp
```

Утилита может работать в нескольких режимах. Начнем с простого режима командной строки. В этом режиме формат команды проверки доступности портов на удаленном сервере следующий:

**PortQry-n <name> [options]**

где основные опции команды:

- n <name>** — имя или IP-адрес компьютера для запроса. Это единственный обязательный параметр для режима командной строки. Это значение не может содержать пробелы;
- e <port\_number>** — порт для запроса. Может иметь значение от 1 до 65535, по умолчанию используется 80;
- p <protocol>** — протокол запроса. Это имеет значение TCP, UDP или BOTH, по умолчанию используется TCP.

Для проверки доступности DNS-сервера 8.8.8.8 по 53 порту UDP выполним следующую команду:

**PortQry-n 8.8.8.8 -e 53 -p UDP**

```
Administrator: Windows PowerShell
PS C:\PortQryV2> .\PortQry.exe -n 8.8.8.8 -e 53 -p UDP
Querying target system called:
8.8.8.8
Attempting to resolve IP address to a name...
IP address resolved to dns.google
querying...
UDP port 53 (domain service): LISTENING or FILTERED
Sending DNS query to UDP port 53...
DNS query timed out
PS C:\PortQryV2>
```

По результату запроса для указанного порта PortQry возвращает одно из трех состояний:

- **Listening** – указанный порт доступен и принимает входящие подключения, ответ от него получен;
- **Not Listening** – по указанному адресу нет процесса (службы и т.п.), который принимает подключения на заданном порту. При проверке доступности ресурса по ICMP утилита PortQry получила ответ Destination Unreachable с кодом Port Unreachable;
- **Filtered** – утилита PortQry не получила ответа от указанного порта, либо ответ был отфильтрован. Т.е. на целевой системе указанный порт никто не слушает, либо доступ к нему ограничен, например файерволом. По умолчанию PortQry запрашивает TCP-порт три раза, а UDP-порт один раз, прежде чем возвращает ответ **Filtered**.

В нашем случае ответ Filtered, т.е. удаленный ресурс доступен, но по указанному порту не отвечает. Что собственно и подтверждает наши подозрения 😊

Добавив ключ -i можно запустить утилиту в интерактивном режиме:

```
PortQry-i -n 8.8.8.8 -e 53 -p UDP
```

и выполнить запрос командой

```
q
```

```
Administrator: Windows PowerShell

PS C:\PortQryV2> .\PortQry.exe -i -n 8.8.8.8 -e 53 -p udp

PortQry Interactive Mode

Type 'help' for a list of commands

Default Node: 8.8.8.8

Current option values:
  end port= 53
  protocol= UDP
  source port= 0 (ephemeral)
> q

resolving service name using local services file...
UDP port resolved to the 'domain' service

IP address resolved to dns.google

querying...

UDP port 53 (domain service): LISTENING or FILTERED

Sending DNS query to UDP port 53...

DNS query timed out

> _
```

В интерактивном режиме можно на лету менять параметры запроса. Для примера изменим адрес сервера:

`node 77.88.8.8`

протокол:

`set protocol=both`

и еще раз отправим запрос:

`q`

Как видите, этот DNS-сервер доступен по всем портам.

Ну и для выхода из интерактивного режима надо набрать:

`exit`

```
Administrator: Windows PowerShell
> node 77.88.8.8
Default Node: 77.88.8.8
>
> set protocol=both
> q

resolving service name using local services file...
TCP port resolved to the 'domain' service

IP address resolved to dns.yandex.ru

querying...

TCP port 53 (domain service): LISTENING

UDP port resolved to the 'domain' service

IP address resolved to dns.yandex.ru

querying...

UDP port 53 (domain service): LISTENING

> exit

exiting PortQry Interactive Mode...
```

В итоге диагностика успешно проведена, проблемы выявлена и устранена. А я еще немного расскажу об утилите.

При работе в интерактивном режиме можно использовать готовые профили служб. Для примера проверим наш многострадальный DNS-сервер. Запустим утилиту в интерактивном режиме:

**PortQry-i**

укажем адрес сервера:

**node 8.8.8.8**

и запустим проверку с помощью предустановленного профиля DNS:

**q dns**

Эта команда проверяет указанный сервер по 53 порту TCP и UDP и эквивалентна команде:

**PortQry -n 8.8.8.8 -p both -e 53**

```
Administrator: Windows PowerShell

PS C:\PortQryV2> .\PortQry.exe -i

PortQry Interactive Mode

Type 'help' for a list of commands

Default Node: 127.0.0.1

Current option values:
  end port=      80
  protocol=     TCP
  source port= 0 (ephemeral)
> node 8.8.8.8
Default Node: 8.8.8.8

>

> q dns

resolving service name using local services file...
UDP port resolved to the 'domain' service

IP address resolved to dns.google

querying...

UDP port 53 (domain service): LISTENING or FILTERED

Sending DNS query to UDP port 53...

UDP port 53 is LISTENING

>

resolving service name using local services file...
TCP port resolved to the 'domain' service

IP address resolved to dns.google

querying...

TCP port 53 (domain service): LISTENING

> ■
```

Профили есть для следующих служб.

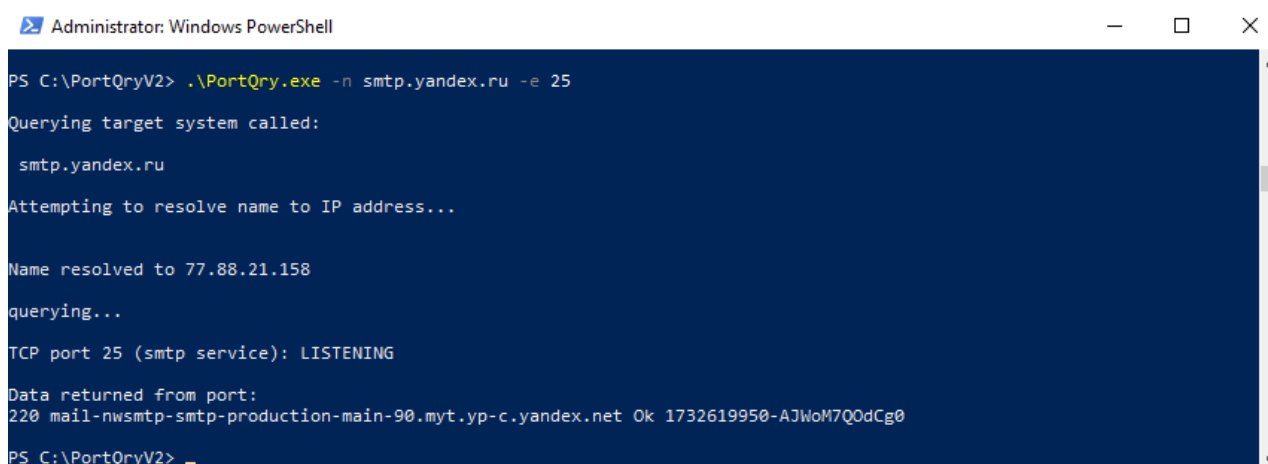
Профиль	Порты для запроса
dns	TCP-порт 53, UDP-порт 53
ftp	TCP-порт 21
imap	TCP-порт 143
ipsec	UDP-порт 500
isa	TCP-порт 1745, UDP-порт 1745
ldap	TCP-порт 389, UDP-порт 389
l2tp	UDP-порт 1701
mail	TCP-порты 25, 110 и 143
pop3	TCP-порт 110
rpc	TCP-порт 135, UDP-порт 135
smtp	TCP-порт 25
snmp	UDP-порт 161



Профиль	Порты для запроса
sql	TCP-порт 1433, UDP-порт 1434
tftp	UDP-порт 69

Также PortQry умеет не просто стучаться по указанным портам, но и производить расширенную диагностику для некоторых сетевых служб. Например при отправке запроса на SMTP-сервер она не только покажет состояние порта, но и выведет приветственный баннер:

`PortQry-n smtp.yandex.ru -e 25`



```

Administrator: Windows PowerShell

PS C:\PortQryV2> .\PortQry.exe -n smtp.yandex.ru -e 25
Querying target system called:
    smtp.yandex.ru
Attempting to resolve name to IP address...
Name resolved to 77.88.21.158
querying...
TCP port 25 (smtp service): LISTENING
Data returned from port:
220 mail-nwsmtp-smtp-production-main-90.myt.yandex.net Ok 1732619950-AJWoM7QOdCg0
PS C:\PortQryV2>

```

А при обращении к контроллеру домена по 389 порту отправит LDAP-запрос:

`PortQry -n srv01.test.local -e 389 -p udp`

```
Administrator: Windows PowerShell

PS C:\PortQryV2> .\PortQry.exe -n srv01.test.local -e 389 -p udp

Querying target system called:

    srv01.test.local

Attempting to resolve name to IP address...

Name resolved to 192.168.147.160

querying...

UDP port 389 (unknown service): LISTENING or FILTERED

Using ephemeral source port
Sending LDAP query to UDP port 389...

LDAP query response:

domainFunctionality: 7
forestFunctionality: 7
domainControllerFunctionality: 7
rootDomainNamingContext: DC=test,DC=local
ldapServiceName: test.local:srv01$@TEST.LOCAL
isGlobalCatalogReady: TRUE
supportedSASLMechanisms: GSSAPI
supportedLDAPVersion: 3
supportedLDAPPolicies: MaxPoolThreads
supportedControl: 1.2.840.113556.1.4.319
supportedCapabilities: 1.2.840.113556.1.4.800
subschemaSubentry: CN=Aggregate,CN=Schema,CN=Configuration,DC=test,DC=local
serverName: CN=SRV01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=test,DC=local
schemaNamingContext: CN=Schema,CN=Configuration,DC=test,DC=local
namingContexts: DC=test,DC=local
isSynchronized: TRUE
highestCommittedUSN: 65813
dsServiceName: CN=NTDS Settings,CN=SRV01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=test,DC=local
dnsHostName: SRV01.test.local
defaultNamingContext: DC=test,DC=local

currentdate: 11/26/2024 11:22:13 (unadjusted GMT)
configurationNamingContext: CN=Configuration,DC=test,DC=local

===== End of LDAP query response =====

UDP port 389 is LISTENING

PS C:\PortQryV2>
```

Еще в PortQry есть локальный режим, с помощью которого можно диагностировать проблемы на локальной системе. Например следующая команда выведет список всех открытых портов на локальном компьютере:

**PortQry -local**

```
Administrator: Windows PowerShell

PS C:\PortQryV2> .\PortQry.exe -local

Processing local system's ports...

Port to process mappings unavailable

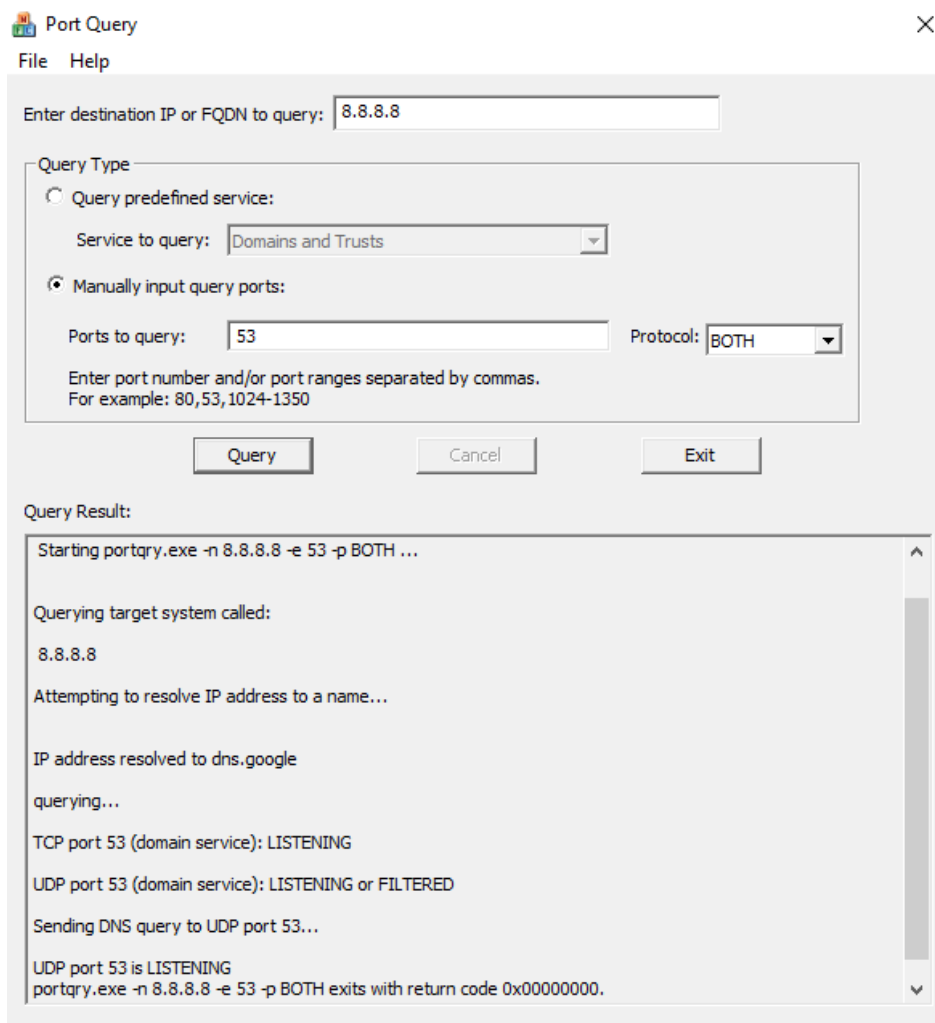
TCP/UDP Port Usage

2552 active ports found

Port      Local IP      State      Remote IP:Port
TCP 53     127.0.0.1     LISTENING  0.0.0.0:0
TCP 53     192.168.147.156 LISTENING  0.0.0.0:0
UDP 53     127.0.0.1     *:*
UDP 53     192.168.147.156 *:*
TCP 88     0.0.0.0       LISTENING  0.0.0.0:0
UDP 88     192.168.147.156 *:*
UDP 123    0.0.0.0
TCP 135    0.0.0.0       LISTENING  0.0.0.0:0
UDP 137    192.168.147.156 *:*
UDP 138    192.168.147.156 *:*
TCP 139    192.168.147.156 LISTENING  0.0.0.0:0
TCP 389    0.0.0.0       LISTENING  0.0.0.0:0
UDP 389    0.0.0.0       *:*
TCP 445    0.0.0.0       LISTENING  0.0.0.0:0
TCP 464    0.0.0.0       LISTENING  0.0.0.0:0
UDP 464    192.168.147.156 *:*
UDP 500    0.0.0.0       *:*
TCP 593    0.0.0.0       LISTENING  0.0.0.0:0
TCP 636    0.0.0.0       LISTENING  0.0.0.0:0
TCP 1032   192.168.147.156 ESTABLISHED 172.64.41.4:443
```

**Примечание.** В документации заявлено, что с помощью параметра `-wport` можно выполнить проверку состояния указанного порта, а параметр `-wpid` выводит состояние всех портов, связанных с указанным процессом на локальном хосте. Но тут есть нюанс — ни на одной системе у меня эти параметры не сработали, при попытке выполнить команду я стабильно получал сообщение "Port to process mapping is not supported on this system".

Ну и для тех, кто не любит набирать команды вручную, есть PortQryUI — вариант PortQry с графической оболочкой. Устанавливается так же, как и консольный вариант утилиты, при запуске открывается окошко, в котором надо указать имя или IP адрес удаленного сервера, порт и протокол для проверки. Также можно выбрать из списка предустановленные профиль службы.



В графическом режиме утилита возвращает следующие коды состояния:

- **0 (0x00000000)** – соединении успешно установлено, порт доступен. Аналогично состоянию **Listening**;
- **1 (0x00000001)** – порт недоступен. Нет процесса, который принимает подключения на заданном порту, либо хост недоступен. Аналогично состоянию **Not Listening**;
- **2 (0x00000002)** – не получен ответ от указанного порта, либо ответ был отфильтрован. Аналогично состоянию **Filtered**.

Как видите, утилита PortQry одна может заменить кучу инструментов для диагностики сетевых проблем, а в случае с UDP она вообще незаменима. На сайте Microsoft можно найти [подробную документацию](#) и [примеры использования](#) утилиты. Так что крайне рекомендую.

Комментарии

Пока нет комментариев.

Ответить