

# Forum:[Hack Forums](#)

Thread: [Pscan.exe ~ IP Address Port Scanner ~ Ready to Use Program + Source Code](#)

Hackers - March 21, 2024, 5:53 am

PSCAN.EXE

Your portable IP scanning tool

Pscan is a small utility tool that can scan for open ports on any IP address. You can look for some common, default ports, or you can enter your own range to search for.

Download the Program

Choose 1 link

Code <https://mega.nz/file/wnEiFaSQ#YnMdbi6ZwyJPQ1vNNBNPZmGOCaLBikOCU0VLRLDiiw>

<https://www.mediafire.com/file/7scst8scv7cfdi/Pscan-Python-v0.0.1-nodes.exe/file>

This is my first time using Pyinstaller for a release like this in years, so please let me know if you have any issues with it working.

Get the sauce!

Code

```
import socket
```

```
def is_port_open(ip, port):
```

```
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
    sock.settimeout(1) # Timeout of 1 second
```

```
    try:
```

```
        sock.connect((ip, port))
```

```
    except socket.error:
```

```
        return False
```

```
    return True
```

```
def scan_ports(ip, mode='common', min_port=1, max_port=65535, common_ports=[21, 22, 23, 25, 80, 110, 443]):
```

```
    open_ports = []
```

```
    if mode == 'r':
```

```
        # Scan the specified range of ports
```

```
        for port in range(min_port, max_port + 1):
```

```
            if is_port_open(ip, port):
```

```
                open_ports.append(port)
```

```
    else:
```

```
        # Scan common ports
```

```
        for port in common_ports:
```

```
            if is_port_open(ip, port):
```

```
                open_ports.append(port)
```

```
return open_ports
```

```
# Example usage
```

```
if __name__ == "__main__":
```

```
    end = False
```

```
    while(end == False):
```

```
        target_ip = input("Enter IP address to scan (or enter Q to quit): ")
```

```
        if(target_ip == "Q" or target_ip == "q"):
```

```
            end = True
```

```
            break
```

```
        mode = input("Enter scan mode ('d' or 'r'). Type in d (default / common ports) or r (range),  
press enter: ").lower()
```

```
        if mode == 'r':
```

```
            min_port = int(input("Enter minimum port number: "))
```

```
            max_port = int(input("Enter maximum port number: "))
```

```
            print(f"Scanning range {min_port}-{max_port}...")
```

```
            open_ports = scan_ports(target_ip, mode, min_port, max_port)
```

```
        elif(mode == "Q" or mode == "q"):
```

```
            end = True
```

```
            break
```

```
        else:
```

```
            print("Scanning common ports...")
```

```
            open_ports = scan_ports(target_ip)
```

```
        print("Open ports:", open_ports)
```

```
        print("\n\n\n")
```

Compile It!

To compile into an exe program, you can use Pyinstaller:

Code pyinstaller --onefile pscan.py

Just run that command from the same directory as your file and it will convert it to a Windows program in the following relative path (or similar):

Code dist/Pscan.exe

Deja Vu?

There are two versions of Pscan originally: the C++ version, and the Python (using Pyinstaller) version.

The Python version, for this release in this thread, was modified from the original to include a loop (to scan more than 1 IP per terminal session) and the addition of the 'Q' entry to quit the program.

If you want to learn more about how they work or make your own custom program, check out the original [port scanning hacking tutorial](#).

**terrytickle** - March 21, 2024, 10:42 am

So, I used this, and found [22] ports.

But it didn't tell me what they were? Is this not pretty useless? lol

**Hackers** - March 21, 2024, 10:47 am

(Mar 21, 2024 10:42 AM)terrytickle Wrote: [\[CITE\]](#) So, I used this, and found [22] ports.

But it didn't tell me what they were? Is this not pretty useless? lol

This is your first post on the forum? Pretty ballsy to be calling something useless? lol

You are seeing the response that port 22 is open.  
Open Ports: [22]  
that means there is 1 port open, and 22 is the port.

Usually I try to be as nice as possible to new users, but your first post calling my software useless and being incorrect is not a great first impression...

**so9or** - March 21, 2024, 3:59 pm  
so nice buddy , there are many errors in your programming

**Hackers** - March 21, 2024, 5:10 pm  
(Mar 21, 2024 03:59 PM)so9or Wrote: [\[CITE\]](#) so nice buddy , there are many errors in your programming  
I put a disclaimer saying this.I don't usually use pyinstaller.So if you have any problems just put the bugs here and I'll get them fixed thanks

**terrytickle** - March 23, 2024, 8:39 am  
[quote="nodes" pid='62309883' dateline='1711043264']

Sorry. I assumed your program was telling me 22 ports were open.  
  
Maybe you should change it, to output:

Found Ports: X  
Port: [22], [55], etc?

Because I thought you found 22 open ports, but didn't tell me what they were. My bad.

**Redemption** - March 23, 2024, 9:11 am  
(Mar 23, 2024 08:39 AM)terrytickle Wrote: [\[CITE\]](#) [quote="nodes" pid='62309883' dateline='1711043264']

Sorry. I assumed your program was telling me 22 ports were open.  
  
Maybe you should change it, to output:

Found Ports: X  
Port: [22], [55], etc?

Because I thought you found 22 open ports, but didn't tell me what they were. My bad.  
The output doesn't need changing, it does exactly what it's supposed to do. I'm not trying to be rude, but based on this post and your previous one, it's pretty clear you're unfamiliar with ports. Here's a fairly short [YouTube video](#) that explains what ports are, and how they work. Once you know what ports are, the output of the script is easy to understand.

**terrytickle** - March 23, 2024, 9:25 am  
Not really.

I know what a port is, and I was expecting more than one port to be open from a large search I did with this tool.

So when the output was

Opened Ports: [22] , I assumed your script broke and didn't show me the actual open ports.

Since you had no other replies on your thread, it looks like a brand new script and noone else has tried it.

Adding an extra line of code:

A total of X ports were found open:  
[22]

would be much better.

KIND REGARDS

**Hackers** - March 23, 2024, 9:36 am

(Mar 23, 2024 09:25 AM)terrytickle Wrote: [\[CITE\]](#) Not really.

I know what a port is, and I was expecting more than one port to be open from a large search I did with this tool.

So when the output was

Opened Ports: [22] , I assumed your script broke and didn't show me the actual open ports.

Since you had no other replies on your thread, it looks like a brand new script and noone else has tried it.

Adding an extra line of code:

A total of X ports were found open:  
[22]

would be much better.

KIND REGARDS

I understand what you are saying.

The reason for [ ] is because that's an array, so usually it says Open Ports: [22, 80, 443] --- it's more apparent when there is more than 1.

If I do another release I'll add this in, otherwise just write a line like this one to say the number of ports w/ the length of the open\_ports list.

```
Code print("Open ports:", open_ports)
```

**G.P.D.** - April 27, 2024, 3:58 am

Hi OP, thank you for sharing your script

While I prefer to use nmap for port scanning, I appreciate the hands-on coding experience your

script provides

I'd like to give some feedback: the one-second delay between scanning each port really impacts the duration of a scan, especially for a wide range (e.g. 20 to 4444). You might consider using futures for parallel execution to enhance efficiency.

Here is an example of how you could implement this:

```
Code import socket
import concurrent.futures
```

```
def is_port_open(ip, port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.settimeout(1)
    try:
        sock.connect((ip, port))
    except socket.error:
        return False
    return True
```

```
def scan_ports(ip, mode='common', min_port=1, max_port=65535, common_ports=[21, 22, 23, 25,
80, 110, 443]):
    open_ports = []
    if mode == 'r':
        with concurrent.futures.ThreadPoolExecutor() as executor:
            port_range = range(min_port, max_port + 1)
            results = [executor.submit(is_port_open, ip, port) for port in port_range]
            for future in concurrent.futures.as_completed(results):
                if future.result():
                    open_ports.append(future.result())
    else:
        with concurrent.futures.ThreadPoolExecutor() as executor:
            results = [executor.submit(is_port_open, ip, port) for port in common_ports]
            for future in concurrent.futures.as_completed(results):
                if future.result():
                    open_ports.append(future.result())
    return open_ports
```

```
if __name__ == "__main__":
    end = False
    while(end == False):
        target_ip = input("Enter IP address to scan (or enter Q to quit): ")
        if(target_ip == "Q" or target_ip == "q"):
            end = True
            break
        mode = input("Enter scan mode ('d' or 'r'). Type in d (default / common ports) or r (range),
press enter: ").lower()
        if mode == 'r':
            min_port = int(input("Enter minimum port number: "))
            max_port = int(input("Enter maximum port number: "))
            print(f"Scanning range {min_port}-{max_port}...")
            open_ports = scan_ports(target_ip, mode, min_port, max_port)
            elif(mode == "Q" or mode == "q"):
```

```
end = True
break
else:
    print("Scanning common ports...")
    open_ports = scan_ports(target_ip)
    print("Open ports:", open_ports)
    print("\n\n\n")
```

**Hackers** - April 28, 2024, 9:45 pm

(Apr 27, 2024 03:58 AM)G.P.D. Wrote: [\[CITE\]](#) Hi OP, thank you for sharing your script

While I prefer to use nmap for port scanning, I appreciate the hands-on coding experience your script provides

I'd like to give some feedback: the one-second delay between scanning each port really impacts the duration of a scan, especially for a wide range (e.g. 20 to 4444). You might consider using futures for parallel execution to enhance efficiency.

thanks, this is really good to look into. The big ranges are definitely impacted. I'm going to rethink this and see what I can do to release a new version. Good point - cheers!

**thebigboss** - April 30, 2024, 3:46 am

Very useful! Thanks for sharing

**Hackers** - April 30, 2024, 4:44 am

(Apr 30, 2024 03:46 AM)thebigboss Wrote: [\[CITE\]](#) Very useful! Thanks for sharing

No problem, glad you like it !

**Vagabundo Legendario** - July 12, 2024, 4:27 pm

(Mar 21, 2024 10:42 AM)terrytickle Wrote: [\[CITE\]](#) So, I used this, and found [22] ports.

But it didn't tell me what they were? Is this not pretty useless? lol  
the earth is not flat man

**longchen\_1562** - July 15, 2024, 1:04 am

Can you help me with web research?

**Ivanov\_1** - July 19, 2024, 8:21 am

Nodes i appreciate the initiative of creating/sharing the scanner. Please don't let some stupid comments bother you,keep up the great work as always.

**Hackers** - July 19, 2024, 8:37 am

(Jul 19, 2024 08:21 AM)Ivanov\_1 Wrote: [\[CITE\]](#) Nodes i appreciate the initiative of creating/sharing the scanner. Please don't let some stupid comments bother you,keep up the great work as always.  
thanks! I do not let them get me down.

I think sometimes people respond to these types of releases and they want to nitpick or find ways to improve it. This can be tough because sometimes I literally just want to make something for the sake of making something. Maybe I have 30 mins, maybe I have 60, maybe I have 2 hours. For me, if I am making something for fun, I either release it as-is or there's a good chance I will never return to it.

- that said, Pscan works fine on my side and I've used it a few times now myself.

If anyone has any issues , just let me know, but keep in mind that I am not going to update this again unless it's literally broken.

**memory** - July 19, 2024, 9:19 am

Always with the nifty releases. Cool share, try not to let others get you down lol you're always taking heat hahah

**Hackers** - July 19, 2024, 12:26 pm

(Jul 19, 2024 09:19 AM)memory Wrote: [\[CITE\]](#) Always with the nifty releases. Cool share, try not to let others get you down lol you're always taking heat hahah

lol it's okay, I don't think anyone reputable has really bashed me so it doesn't feel that way even if it looks that way.

**lasvegas** - July 30, 2024, 7:27 pm

(Mar 21, 2024 10:42 AM)terrytickle Wrote: [\[CITE\]](#) So, I used this, and found [22] ports.

But it didn't tell me what they were? Is this not pretty useless? lol

It means port 22 is open, most likely being SSH

(Mar 21, 2024 05:53 AM)nodes Wrote: [\[CITE\]](#) PSCAN.EXE

Your portable IP scanning tool

Pscan is a small utility tool that can scan for open ports on any IP address. You can look for some common, default ports, or you can enter your own range to search for.

Download the Program

Choose 1 link

Code <https://mega.nz/file/wnEiFaSQ#YnMdbi6ZwyJPQ1vNNBNPZmGOCaLBikOCU0VLRLDiiw>

<https://www.mediafire.com/file/7scst8scv7cfdii/Pscan-Python-v0.0.1-nodes.exe/file>

This is my first time using Pyinstaller for a release like this in years, so please let me know if you have any issues with it working.

Get the sauce!

Code import socket

```
def is_port_open(ip, port):
```

```
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
    sock.settimeout(1) # Timeout of 1 second
```

```
    try:
```

```
        sock.connect((ip, port))
```

```
    except socket.error:
```

```
        return False
```

```
    return True
```

```
def scan_ports(ip, mode='common', min_port=1, max_port=65535, common_ports=[21, 22, 23, 25, 80, 110, 443]):
```

```
    open_ports = []
```

```
    if mode == 'r':
```

```
        # Scan the specified range of ports
```

```
        for port in range(min_port, max_port + 1):
```

```
            if is_port_open(ip, port):
```

```

        open_ports.append(port)
    else:
        # Scan common ports
        for port in common_ports:
            if is_port_open(ip, port):
                open_ports.append(port)
    return open_ports

# Example usage
if __name__ == "__main__":
    end = False
    while(end == False):
        target_ip = input("Enter IP address to scan (or enter Q to quit): ")
        if(target_ip == "Q" or target_ip == "q"):
            end = True
            break
        mode = input("Enter scan mode ('d' or 'r'). Type in d (default / common ports) or r (range),
press enter: ").lower()
        if mode == 'r':
            min_port = int(input("Enter minimum port number: "))
            max_port = int(input("Enter maximum port number: "))
            print(f"Scanning range {min_port}-{max_port}...")
            open_ports = scan_ports(target_ip, mode, min_port, max_port)
        elif(mode == "Q" or mode == "q"):
            end = True
            break
        else:
            print("Scanning common ports...")
            open_ports = scan_ports(target_ip)
        print("Open ports:", open_ports)
        print("\n\n\n")

```

### Compile It!

To compile into an exe program, you can use Pyinstaller:

Code `pyinstaller --onefile pscan.py`

Just run that command from the same directory as your file and it will convert it to a Windows program in the following relative path (or similar):

Code `dist/Pscan.exe`

### Deja Vu?

There are two versions of Pscan originally: the C++ version, and the Python (using Pyinstaller) version.

The Python version, for this release in this thread, was modified from the original to include a loop (to scan more than 1 IP per terminal session) and the addition of the 'Q' entry to quit the program.

If you want to learn more about how they work or make your own custom program, check out the original [port scanning hacking tutorial](#).

Suggestion: you can replace

Code `print("Open ports:", open_ports)` With:

Code `print("Open ports:", *open_ports, sep=', ')` This will print the array of ports more nicely (without the brackets)

Hackers - July 30, 2024, 10:13 pm

(Jul 30, 2024 07:27 PM)lasvegas Wrote: [\[CITE\]](#) Suggestion: you can replace  
Code print("Open ports:", open\_ports)With:  
Code print("Open ports:", \*open\_ports, sep=', ')This will print the array of ports more nicely (without  
the brackets)  
Smart, thanks for the idea. I see it caused some confusion for the user before obviously :)

lasvegas - July 31, 2024, 7:06 am

(Jul 30, 2024 10:13 PM)nodes Wrote: [\[CITE\]](#) Smart, thanks for the idea. I see it caused some  
confusion for the user before obviously :)  
no problem! you can also make some request to understand what is being hosted on that port (ssh,  
http, ftp etc..)

<https://github.com/nmap/nmap>

Hackers - July 31, 2024, 9:10 am

(Jul 31, 2024 07:06 AM)lasvegas Wrote: [\[CITE\]](#) no problem! you can also make some request to  
understand what is being hosted on that port (ssh, http, ftp etc..)

<https://github.com/nmap/nmap>

Yes, definitely better to scan for real protocols than to assume each port is set to default

I'm not sure if I'll come back to this program ever but definitely something I would want to add.  
Great suggestion

Stanley - November 19, 2024, 10:06 am

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page  
and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP  
(Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for  
each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first  
post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,  
Stanley

Radigast - December 1, 2024, 11:31 am

Great Utility Concept. I may have some suggestions to help you out on your journey.

Port scanning is fundamental for both ethical hacking and network diagnostics. You've done well to  
include the option for scanning common ports or a user-defined range. The simplicity of this tool  
makes it accessible to beginners while remaining practical for advanced users. Again, Great work.  
Best idea I have come across all week.

Enhance Output Details

Consider providing more information about the open ports. For example, associating each port with  
its common service (e.g., port 80 -> HTTP, port 22 -> SSH) would make the output more  
informative. You could add:

```
port_services = {21: 'FTP', 22: 'SSH', 23: 'Telnet', 25: 'SMTP', 80: 'HTTP', 110: 'POP3', 443:  
'HTTPS'}
```

Then display the service name in the results.

----

#### Add Error Handling

You're already handling timeouts, but consider catching exceptions for invalid IP addresses or permission errors. For instance:

```
try:
    socket.inet_aton(ip) # Validate IP format
except socket.error:
    print("Invalid IP address. Please try again.")
    continue
```

#### Multi-Threading for Speed

Scanning ports sequentially can be slow, especially for larger ranges. Implementing threading or asynchronous scanning could significantly improve performance. Python's `concurrent.futures.ThreadPoolExecutor` is perfect for this.

Example: `from concurrent.futures import ThreadPoolExecutor`

```
def scan_ports_async(ip, ports):
    open_ports = []
    with ThreadPoolExecutor(max_workers=100) as executor:
        results = executor.map(lambda p: (p, is_port_open(ip, p)), ports)
        open_ports = [p for p, is_open in results if is_open]
    return open_ports
```

#### Interactive Enhancements

Add a feature to save scan results to a file or display results in a user-friendly format, like a table. This makes it easier to analyze findings later.

#### Make It Cross-Platform

Since Pyinstaller generates an .exe for Windows, consider building versions for other platforms (Linux, macOS). Tools like `py2app` for macOS or `cx_Freeze` can help with this.

Future Improvements: If you're considering evolving this project further, here are some ideas:

1. Add a ping sweep feature to detect live hosts in a subnet.
2. Integrate with a small web UI using a framework like Flask for remote access.
3. Incorporate advanced scanning options like TCP/UDP flags or service fingerprinting.

Overall, solid work my fellow digital nomad!! Let me know if you'd like help implementing any of these ideas or feedback. Keep pushing this project forward—simple tools like this are often the most versatile and widely appreciated. ?

Radigast

HackThePlanet

Hackers - March 12, 2026, 9:02 pm

This program still works and is easy to use