

Forum:[Hack Forums](#)

Thread: [python scanner source code](#)

Dcom_3xp10it - June 28, 2016, 6:28 pm

Hi, I try to change the code found, for it then run a scan ip ranks while checking the predefined ports in the code is open, if open port on an IP address during the scan, save to a file IP txt

```
Code #!/usr/bin/env python
import socket
import subprocess
import sys
from datetime import datetime
```

```
# Clear the screen
subprocess.call('cls', shell=True)
```

```
# Ask for input
remoteServer = raw_input("Enter a remote host to scan: ")
remoteServerIP = socket.gethostbyname(remoteServer)
```

```
# Print a nice banner with information on which host we are about to scan
print "-" * 60
print "Please wait, scanning remote host", remoteServerIP
print "-" * 60
```

```
# Check what time the scan started
t1 = datetime.now()
```

```
# Using the range function to specify ports (here it will scans all ports between 1 and 1024)
```

```
# We also put in some error handling for catching errors
```

```
try:
    for port in range(1,1000):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        result = sock.connect_ex((remoteServerIP, port))
        if result == 0:
            print "Port {}:    Open".format(port)
            sock.close()
```

```
except KeyboardInterrupt:
    print "You pressed Ctrl+C"
    sys.exit()
```

```
except socket.gaierror:
    print 'Hostname could not be resolved. Exiting'
    sys.exit()
```

```
except socket.error:
```

```
print "Couldn't connect to server"  
sys.exit()
```

```
# Check the time again  
t2 = datetime.now()
```

```
# Calculate total time; subtract start time from end time  
total = t2 - t1
```

```
# Print the total time on screen  
print '\n'  
print '-' * 60  
print "Total scan time: ", total  
print '-' * 60
```

Goodies - June 28, 2016, 7:24 pm

You're spawning a new process for each scan? Good lord... a TCP syn request is very cheap in terms of CPU usage. Incredibly cheap. The vast majority of the time is spent waiting for the response. For this reason you should not be using processes or threads. Especially not threads. You should be using asynchronous programming.

lhabZ - June 29, 2016, 3:30 am

thanks for sharing bro :D
!

dreadpixel - July 8, 2016, 3:25 pm

Thanks for this educational post