

Forum:[Hack Forums](#)

Thread: [\[tut\] creating a TCP port scanner in Visual Basic.net](#)

**** - April 18, 2011, 7:40 pm

Today i will demonstrate how to create your own port scanner in Visual basic.net

Basically, it will preform the same function as n-map (albeit with much less features)

How will this work?

Basically, this port scanner will try to create a TCP connection to every port within the specified port range. If the connection can be successfully established, than the port will be listed as open, if the connection cannot be opened, than the port will be listed as closed.

Basically, this is how every port scanner works. However, better port scanners like n-map can scan using more methods than just attempting to establish a tcp connection. fur example the syn scan.

unfortunately, this port scanner will list every port that it cannot establish a connection with as "closed". this might not be accurate, as there can be many reasons why a connection cannot be established.

creating the port scanner

first of all, create a new console application, and delete all the code

import the required stuff:

Code Imports System.Net.Sockets

Imports System.Net

Imports System.Threading

Imports System.Textthese imports will allow us to use many much needed methods, like

Tcpconnect()

now the class:

Code Public Class psAPP

Public Shared Sub Main()

End Sub

End Class

and declare the variables

Code Dim portStart As Integer

Dim portEnd As Integer

Dim lngPort As Long

Dim txtHost As String

Dim openPorts As Integer

Dim closedPorts As Integer

the user input:

Code Console.Write("Host: ")

txtHost = Console.ReadLine()

Console.Write("From: ")

```
portStart = Console.ReadLine()
```

```
Console.Write("To: ")
```

```
portEnd = Console.ReadLine()this is to determine the host to scan , and the range of ports to scan
```

scanning the host:

```
Code For lngPort = portStart To portEnd
```

```
Dim myTcpClient As New TcpClient()
```

```
Try
```

```
myTcpClient.Connect(txtHost, lngPort)
```

```
Console.WriteLine("Host: " + txtHost + " : ")
```

```
Console.WriteLine(" Port " + lngPort.ToString() + " : Open :")
```

```
openPorts += 1
```

```
myTcpClient.Close()
```

```
Catch ex As SocketException
```

```
Console.WriteLine("Host: " + txtHost + " : ")
```

```
Console.WriteLine(" Port " + lngPort.ToString() + " : Closed :")
```

```
' Console.WriteLine(ex.Message)
```

```
closedPorts += 1
```

```
End Try
```

```
Next
```

this works by trying to connect to every port within the scan range, and see if a connection can be established or not

output code:

```
Code Console.Write(" " & openPorts.ToString & " open port(s) : ")
```

```
Console.Write(" " & closedPorts.ToString & " closed port(s) : ")
```

```
Console.Beep()
```

```
Console.Write(txtHost + " : " + portStart.ToString + " - " + portEnd.ToString + " : Scanned
```

```
Sucessfully")
```

this is the final code that displays the out come to the user

example code

```
Code Imports System.Net.Sockets
```

```
Imports System.Net
```

```
Imports System.Threading
```

```
Imports System.Text
```

```
Public Class Tester
```

```
Public Shared Sub Main()
```

```
Dim portStart As Integer
```

```
Dim portEnd As Integer
```

```
Dim lngPort As Long
```

```
Dim txtHost As String
```

```
Dim openPorts As Integer
```

```
Dim closedPorts As Integer
```

```
Console.Write("Host: ")
```

```
Console.Write("")
```

```
txtHost = Console.ReadLine()
```

```
Console.Write("From: ")
```

```
Console.Write("")
portStart = Console.ReadLine()
```

```
Console.Write("To: ")
Console.Write("")
portEnd = Console.ReadLine()
```

```
Console.WriteLine("")
```

```
For lngPort = portStart To portEnd
    Dim myTcpClient As New TcpClient()
    Try
        myTcpClient.Connect(txtHost, lngPort)
        Console.WriteLine("Host: " + txtHost + " : ")
        Console.WriteLine(" Port " + lngPort.ToString() + " : Open :")
        openPorts += 1
        myTcpClient.Close()
        Catch ex As SocketException
            Console.WriteLine("Host: " + txtHost + " : ")
            Console.WriteLine(" Port " + lngPort.ToString() + " : Closed :")
            ' Console.WriteLine(ex.Message)
            closedPorts += 1
        End Try
    Next
```

```
Console.WriteLine("")
Console.Write(" " & openPorts.ToString & " open port(s) : ")
```

```
Console.Write(" " & closedPorts.ToString & " closed port(s) : ")
Console.WriteLine("") & vbCrLf)
Console.Beep()
Console.Write(txtHost + " : " + portStart.ToString + " - " + portEnd.ToString + " : Scanned
Sucessfully")
```

```
Console.ReadLine()
End Sub
End Class"and now we are done?
```

SoulZeh - April 18, 2011, 7:49 pm

Post a Screen Shot pleasee
but nice tut <3

********* - April 18, 2011, 7:53 pm

(Apr 18, 2011 07:49 PM)KevvYE Wrote: [\[CITE\]](#) Post a Screen Shot pleasee
but nice tut <3
the result (not created by me though):
[img]http://img38.imageshack.us/img38/6430/finishedporj.png[/img]

Arjun21 - October 18, 2011, 4:26 am

Very nice tutorial man..
Thanks..
Keep posting

