

Forum:[Hack Forums](#)

Thread: [Simple port scanner using nmap](#)

x1n5h3n - November 10, 2019, 7:57 pm

Here is a first version of a simple scanner using nmap and standards libraries.
pytyhon2 and python3 compatible.

The next version will allow to export results in HTML.

Asking review of this first version.

scanner.py

Code

```
import sys
```

```
import argparse
```

```
import nmap
```

```
#Set DEBUG
```

```
DEBUG = False
```

```
#List immutable for args
```

```
input = set()
```

```
def parse_arguments():
```

```
    """Define possible arguments"""
```

```
    desc = """Port scanner outputting its results as an HTML page."""
```

```
    #use argparse to parse args
```

```
    parser = argparse.ArgumentParser(prog='scanner.py', description=desc)
```

```
    parser.add_argument('-o', '--host',
```

```
        metavar='Hostanme',
```

```
        help='host name ex: google.com',
```

```
        nargs='+')
```

```
    parser.add_argument('-i', '--ip',
```

```
        metavar='IP',
```

```
        help='ipv4/ipv6 address ex: 192.168.0.1',
```

```
        nargs='+')
```

```
    parser.add_argument('-c', '--cidr',
```

```
        metavar='CIDR',
```

```
        help='CIDR notation ex: 192.168.0.0/24',
```

```
        nargs='+')
```

```
    parser.add_argument('-f', '--file',
```

```
        metavar='File',
```

```
        help='file containing a target list (host, ip, cidr)',
```

```
        type=argparse.FileType('r'),
```

```
        nargs='+')
```

```
    #parse args
```

```
    args_tolist(parser, parser.parse_args())
```

```
def file_tolist(args):
```

```
    """Words from file to list"""
```

```
    for file in args.file:
```

```

    for line in file:
        for word in line.split():
            input.add(word)

def args_tolist(parser, args):
    """Arguments to list"""
    try:
        if args.host is not None:
            for host in args.host:
                input.add(host)
        if args.ip is not None:
            for ip in args.ip:
                input.add(ip)
        if args.cidr is not None:
            for cidr in args.cidr:
                input.add(cidr)
        if args.file is not None:
            file_tolist(args)
    except:
        print('Input not reconize !\n')
        parser.print_help()
        sys.exit(1)
    if DEBUG:
        print(input)

def scann():
    """Main function"""
    parse_arguments()
    nmap.open(input)

if __name__ == '__main__':
    scann()
nmap.py
Code import parse
import shlex, subprocess

def open(targets):
    """Execute nmap"""
    #nmap options
    options = " -v "
    #make cmd readable by subprocess.Popen
    cmd = shlex.split("sudo nmap" + options + ' '.join(targets))
    #create a nmap process
    process = subprocess.Popen(cmd, stdout=subprocess.PIPE)
    parse.displayprocess(process, targets)
helper.py
Code def numberofip(targets):
    i = 0
    for e in targets:
        if "/" in e:
            (ip, cidr) = e.split("/")
            cidr = int(cidr)
            i += 2**(32-cidr)

```

```

else:
    i += 1
return i

```

```

def progress(count, total, suffix=""):
    """Display a progress bar"""
    #width of the bar
    bar_len = 20
    filled_len = int(round(bar_len * count / float(total)))
    percents = round(100.0 * count / float(total), 1)
    bar = '=' * filled_len + '-' * (bar_len - filled_len)
    print('[%s] %s%%s \t %s\r' % (bar, percents, '%', suffix))

```

parse.py

Code import helper

import re

```

def displayprocess(process, targets):
    """Display process"""
    #get number of scan
    nb = helper.numberofip(targets) + 1
    #progression counter
    i = 0
    #open ports
    p = 0
    #while the process running
    while process.poll() is None:
        #add output line to line
        line = process.stdout.readline()
        #if line contain Starting
        if "Starting" in line:
            helper.progress(i, nb, re.sub(r'\([^\)]*\)', '', line.replace('\n', '')))
        elif "Discovered" in line:
            continue
        elif "Nmap scan report" in line:
            i += 1
            helper.progress(i, nb, "Scanning " + re.sub(r'Nmap scan report for ', '', line.replace('\n', '')))
        elif "All " in line:
            helper.progress(i, nb, "All " + re.search('\d+', line).group() + " scanned ports are closed")
        elif "open" in line:
            p += 1
            helper.progress(i, nb, "Port " + re.search('\d+', line).group() + " open")
        elif "Nmap done" in line:
            i = nb
            helper.progress(i, nb, re.sub(r'Nmap done: ', '', line.replace('\n', '')))
    print("\nResult: %d open port(s) found !" % (p))

```

Can be run like that: `python scanner.py -o google.com -i 192.168.1.1 -c 192.168.1.0/24`

And, do you know how can I deploy it in production in a secure way ?

HFDone - November 11, 2019, 7:43 am

why nmap [Port scanner in Python](#)

x1n5h3n - November 11, 2019, 8:25 am

(Nov 11, 2019 08:43 AM)HFDone Wrote: [\[CITE\]](#) why nmap [Port scanner in Python](#)
To use the power of nmap like getting @Mac, service version, using NSE scripts, etc.

Kallu609 - November 11, 2019, 9:55 am

You can replace the humongous if-clause with a loop.

x1n5h3n - November 11, 2019, 9:59 am

(Nov 11, 2019 10:55 AM)Kallu609 Wrote: [\[CITE\]](#) You can replace the humongous if-clause with a loop.
In helper.py ?

Kallu609 - November 11, 2019, 10:04 am

(Nov 11, 2019 10:59 AM)x1n5h3n Wrote: [\[CITE\]](#) In helper.py ?
Yep, you just need some basic math.

x1n5h3n - November 11, 2019, 10:13 am

(Nov 11, 2019 11:04 AM)Kallu609 Wrote: [\[CITE\]](#) Yep, you just need some basic math.
Thx Kallu609, found a solution !

Code def numberofip(targets):

```
i = 0
for e in targets:
    if "/" in e:
        (ip, cidr) = e.split("/")
        cidr = int(cidr)
        i += 2**(32-cidr)
    else:
        i += 1
return i
```

Kallu609 - November 11, 2019, 10:28 am

(Nov 11, 2019 11:13 AM)x1n5h3n Wrote: [\[CITE\]](#) Thx Kallu609, found a solution !
Great, I also did version of it:

x1n5h3n - November 11, 2019, 10:37 am

[quote='Kallu609' pid='59136765' dateline='1573496904']

Great, I also did version of it: ...

I use else: i += 1 because my input can be:
['192.168.0.1','192.168.0.0/24','google.com']