

Forum:[Hack Forums](#)

Thread: [Basic Traceroute Script](#)

FriendOfRaven - February 20, 2017, 2:27 pm

Introduction

This is my first tutorial on here so I do apologise if it is poorly written. Feel free to correct me on anything I do explain wrongly. It needs to be run as root on *nix as works with raw sockets. If you do not understand any part of this, post all questions publicly so I can answer for everyone else with the same question. Of course it must be mentioned, in any real task use the proper tool, this is purely an exercise to get more comfortable using python and sockets. Perhaps even gaining better knowledge of networking as a result or hopefully peaking interest into the field.

Theory

Packets get passed from your computer, to your router, then forwarded to switches and routers etc on their way to the server you wish to contact. Network administrators often need to find where there are unnecessary delays in networks or find where the packets get interrupted so that they can narrow down the search for broken links in their network.

Some systems won't respond to the request as it can sometimes be used as part of a DDoS attack. In most cases it is highly unlikely to need more than 30 stops on the way so the script uses 30 'hops' as a default limit. The application that this script is a much simpler version of is traceroute on *nix or tracert on windows. to execute them simply type into a command line "traceroute google.com" or "tracert google.com".

Simplified Pseudocode:

www.pastebin.com/ziLpHj0z

Python Script Outline

www.pastebin.com/LEgJuCmF

Add Imports

There are 3 imports used in this script. The first is [socket](#), used to create the network connections used in this script. To find if there is lag in the network, rather than just finding dead links, this script returns the time taken for each packet to find where there is undue lag. To do this the [time](#) must be imported. The final import used is [sys](#), used to handle arguments passed to the script.

www.pastebin.com/WAr24AB4

Take Argument

Using the sys module, you can accept arguments into your python script. Sys allows you to access variables used by the interpreter functions to interact with them. Arguments passed are stored in list 'argv'. argv[0] is the filename of the script, depending on OS it will either be the file name or the full file path. argv[1] is the first variable passed. To find the length of a list in python it is easiest to use "len(sys.argv)". If a variable has been passed to the script the length will be 2, otherwise 1.

www.pastebin.com/qSFrTNjs

Setting up Socket

Sockets that we will use are recv_socket and send_socket. To create a socket you must pass the variables "Address Family", we will stick to ipv4 in this case which is "AF_INET". Then we pass a

variable to state which of the 7 layers of the network you are using, for future reference layer 1 is physical. We will be using level 3 (Network Layer) with "socket.SOCK_RAW". The final variable is the protocol used which for the recv_socket is ICMP.

For the send_socket we use "socket.SOCK_DGRAM" instead of raw so that we can send an UDP packet. We create the variables "curr_addr" and "curr_name". settimeout is set to 5 seconds to make sure the script doesn't get left hanging when a switch or router etc doesn't respond to the packets timing out. Sendto sends an empty packet as content is an empty string in this case.

www.pastebin.com/iPewf3u2

Receive Responses

This script uses a basic try-except. Try is simply listening for a response, simply jumps out with 'pass' if an exception occurs, and finally closes the sockets. I do not see any reason to explain the rest of this segment as it is quite clear that it prints to the console. Any confusion over the print function please [RTFM](#)

<http://pastebin.com/sqZUXDvy>

Timing

Using 'time', we take note of the start time as "start = time.time()*1000" to get the time in millisecond. Once a response has been received we use "finish = time.time()*1000" and subtract the variable start to get the time taken in milliseconds.

Summary

This is the full code with some tidying. The only call I haven't explained this far is "socket.getfqdn(curr_addr)" which stands for "Fully Qualified Domain Name". The forum wouldn't allow me to use code tags instead saying to host any code on pastebin, so I apologise for all the links.

obby - February 20, 2017, 2:34 pm

Im sure this well be helpful to many interested in learning more about networking.

(???)?(???? - February 20, 2017, 2:36 pm

Good detail - few things.

You should use [code] tags so reads don't have to open up each of the links individually. Also, consider updating to Python3. Python 2 is over 8 years old at this point.

FriendOfRaven - February 20, 2017, 2:41 pm

Thank you Obiwankanobi.

I tried using the code tags but when I submitted it, I got an error saying that the firewall blocked the link.

You have a good point regarding the Python version, I will probably upload some more of my stuff written in this version for a little bit as I have a couple of projects about 90% done, but will change to the newer version in a few days. I appreciate the advice Twixxy.

(???)?(???? - February 20, 2017, 2:52 pm

(Feb 20, 2017 03:41 PM)FriendOfRaven Wrote: [\[CITE\]](#) Thank you Obiwankanobi.

I tried using the code tags but when I submitted it, I got an error saying that the firewall blocked the link.

You have a good point regarding the Python version, I will probably upload some more of my stuff written in this version for a little bit as I have a couple of projects about 90% done, but will change to

the newer version in a few days. I appreciate the advice Twixxy.
You may need to wait until you have 25 posts.

captainspalding - February 20, 2017, 3:59 pm

Keep in mind traceroute RTTs are not by themselves always an accurate indicator of where congestion/latency in a path is.

<https://www.youtube.com/watch?v=WL0ZTcfSvB4>

Many reasons why a router would process ICMP responses slower than an other protocols, like TCP/UDP.

FriendOfRaven - February 20, 2017, 4:13 pm

Very good point CaptainSpalding.

I hadn't thought of that, I hope that people can put together the information in that video as well as the tutorial to get good use out of it.

ven0m - February 20, 2017, 7:59 pm

We have pastebin integration on this site. Use the following format if possible.

Code [pastebin]your pastebin link[/pastebin]

FriendOfRaven - February 20, 2017, 11:27 pm

(Feb 20, 2017 08:59 PM)ven0m Wrote: [CITE](#) We have pastebin integration on this site. Use the following format if possible.

Code [pastebin]your pastebin link[/pastebin]

When making the thread I tried with paste instead of pastebin in the tags, just edited the thread and it didn't work with pastebin tags as you used in the quote. Thanks for trying to help though.

Goodies - February 21, 2017, 1:36 am

Nice! Couple of changes I'd male"

1) "while True" is generally not a good idea in programming. You have a base condition at the bottom, so you should put that in your while condition instead of breaking. Or, at the very least, have some kind of boolean flag that is changed to the opposite value when those conditions are met.

2) When you use socket.bind(), don't bind to that port. You cannot guarantee that there is not already something listening on it, and you don't know that it's available for multiple sockets. The OS will choose a port if you simply give it 0 as a parameter. That being said, generally speaking, you use port 1 as the destination port when you're doing this, but that's not a necessity.

3) Line 25 is doing nothing as it is overridden by line 35.

4) Your sockets are not in context managers. There are also other operations being performed before your try/finally block is hit. For example, if you can create recv_socket and send_socket, but you cannot bind it (permissions error or what have you) then the sockets will stay open and the finally block will not be hit. Also, on line 26, you attempt to use the socket without any exception handling whatsoever. Now, in Python3, sockets are context managers, but if you're too stubborn to move to the best version of the language, you can use contextlib. Particularly, `from contextlib import closing` would be helpful in this case.

4) Your string formatting could use some work. Python is a built-in mini language for string formatting This makes it look much better than tab-separated output.

PHP Code

<?php

```
fmt = "{:<4}{:<35}"
```

Now, throughout the code, you can do. for example:

PHP Code

```
<?php
print(fmt.format("TTL", "Host Details", "Time (ms)")) # print the top chart
print(fmt.format(ttl, curr_host, finish - start))
```

ven0m - February 21, 2017, 4:14 am

(Feb 21, 2017 12:27 AM)FriendOfRaven Wrote: [\[CITE\]](#) When making the thread I tried with paste instead of pastebin in the tags, just edited the thread and it didn't work with pastebin tags as you used in the quote. Thanks for trying to help though.

My apology. It is the [paste][/paste] code.

Example:

Code [paste]<http://pastebin.com/ziLpHj0z>[/paste]

Infamia - February 21, 2017, 6:59 am

(Feb 20, 2017 03:36 PM)(?°?°)?(???? Wrote: [\[CITE\]](#) Good detail - few things.

You should use [code] tags so reads don't have to open up each of the links individually. Also, consider updating to Python3. Python 2 is over 8 years old at this point.

Python3 is also 8 years old.

consider staying on python2 until further notice

(?°?°)?(???? - February 21, 2017, 7:10 am

(Feb 21, 2017 07:59 AM)Infamia Wrote: [\[CITE\]](#) Python3 is also 8 years old.

consider staying on python2 until further notice

Yeah... The first version of it was 8 years ago. Last major version was two months ago.

Infamia - February 21, 2017, 7:31 am

(Feb 21, 2017 08:10 AM)(?°?°)?(???? Wrote: [\[CITE\]](#) Yeah... The first version of it was 8 years ago. Last major version was two months ago.

same

(?°?°)?(???? - February 21, 2017, 7:35 am

(Feb 21, 2017 08:31 AM)Infamia Wrote: [\[CITE\]](#)

same

In my post I meant minor, the 6, but yours is irrelevant. That's a micro version. That's like a spelling fix.

Infamia - February 21, 2017, 7:40 am

(Feb 21, 2017 08:35 AM)(?°?°)?(???? Wrote: [\[CITE\]](#) In my post I meant minor, the 6, but yours is irrelevant. That's a micro version. That's like a spelling fix.

touché