

Mini Network Toolkit

Group - 7

Rushil Patel (AU1841008)

Harshil Mehta (AU1841010)

Project Definition

- Mini Network Toolkit is an application that allows its users to perform some network related functions and monitoring activity.
- The point of such a mini toolkit is to ease the pain of managing the network administrators so that they would not need large and complex softwares just to perform some basic housekeeping.
- Some features of our toolkit are:
 - Fetching the system information.
 - Fetching the Interface Table and the Routing table of the machine.
 - Viewing the connections made by internet sockets and/or unix sockets
 - Capturing and viewing ICMP, TCP and UDP packets
 - Testing whether an IP address or a set of IP addresses are active or unreachable
 - Tracing the path taken by a packet during its journey to the provided machine.

Stepwise approach to solve a problem #1

- 1) To build a simple network toolkit, we have used *python-tkinter* to create its Graphical User Interface.
 - a) For these we have used functionalities / modules such as creating frames, scroll application & other formatting techniques too.
 - b) Proper modular functions (new files) are made to process different functionalities.
- 2) *ViewSystemDetails()* functionality calls the OS from the python environment and dumps the expected output into a data file.
 - a) Now, we create a separate frame / window using tkinter and extract the data from the file.
 - b) Use of proper delimiter's to show the output in proper format.
 - c) The command used here is *uname* with different switches.

Stepwise approach to solve a problem #2

- 1) *KernelInterfaceTable()* and *kernelRoutingTable()* shows the interface table for connecting to the node using *netstat* command with “-i & -r”.0
 - a) Use of tables, scrolls, frames was done to provide proper interfacing to the users.
- 2) *ConnectionDetails()* extracts Packet Connection information and then it is read, formatted and appropriately filtered by the tkinter facility.
 - a) The command used here to view packet connection is ‘ss’.
- 3) *PacketSniffer()* sniffs the network and shows information regarding the following packets.
 - a) ICMP Packets
 - b) TCP Packets
 - c) UDP Packets

We have developed the sniffing functionality ourselves and it does not use any Shell Command like the other functionalities.

Stepwise approach to solve a problem #3

- 1) The *fping()* utility checks the activity of many desired hosts at a single time & provide suitable information regarding the *ping* sent.
 - a) Here, we also have added dynamic addition of hostnames (in form of abc.com / 8.8.8.8), where user can add / remove different hostnames accordingly in the text area given in GUI.
 - b) The final status *alive/unreachable/dead* is displayed with given hostname.
- 2) The final utility is to show the path/hops for the system to connect to any given webservice using *traceroute*.
 - a) Again, here the similar methods of data fetch & feed works for the GUI as well as formatting.

Originality

01

This mini toolkit allows you to access multiple network tracking commands

- Check network statistics even for multiple targets
- Monitoring network through packet capture
- Info about Hops / Path through traceroute & routing information (Data Forwarding)

02

It also allows users to simultaneously check system/connection details

- It helps users to check its active ports & its interface info.
- Reads the specification of the system

03

Integrating the all above mentioned network related commands for feasible use

- Common usage for diverse networking commands
- GUI integrated facility for better user perspective
- Implemented python sniffer to read data on an open socket with proper filters & unpacking.

Tools and References used

Language	Python
GUI Framework	Tkinter (Python)
Other Modules	Socket
	Struct
	Textwrap
	Subprocess
	OS
Shell Commands	fping
	tracert
	netstat
	ss

Conclusion and Future Work

- **Conclusion :**
 - Concluding, we can say that our toolkit works as intended and provides a very useful housekeeping as well as a quite illustrating example for visualising our theory concepts.
 - Obviously, it is not the best software out that can perform such common tasks, but it is definitely usable and could do some good to the network and it's administrator/user, though at a smaller scale and can be used widely in monitoring activities.
- **Future Work :**
 - Can add more validation, making sure the software works as intended.
 - Can add a few more features like capturing different classes of packets and analyse them.
 - A summary feature that shows the statistics of different packets over time.
 - Support for Simple Network Management Protocol (SNMP)
 - Can be scaled to include some control settings

Thank You