TMA PhD school 2014: Middlebox detection via Tracebox

K. Edeline
korian.edeline@ulg.ac.be
Université de Liège

15/03/14

1 Overview

Middleboxes such as firewalls, NAT, proxies, or Deep Packet Inspection play an increasingly important role in various types of IP networks, including enterprise and cellular net- works. Recent studies have shed the light on their impact on real traffic and the complexity of managing them. Network operators and researchers have few tools to understand the impact of those boxes on any path. To address this problem, researchers from ULg and UCL, Belgium proposed a new tool called `tracebox`[1], which is an extension to the widely used `traceroute` tool that is capable of detecting various types of middle- box interference over almost any path. `tracebox` sends IP packets containing TCP segments with different TTL values and analyses the packet encapsulated in the returned ICMP message. Further, as recent routers quote, in the ICMP message, the entire IP packet that they received, `tracebox` is able to detect any modification performed by upstream middleboxes. In addition, `tracebox` can often pinpoint the network hop where the middlebox interference occurs.

In this laboratory session, you will be confronted to different netkit-emulated[1] network topologies. Each of the topologies contains different middleboxes which will affect differently the network traffic. Your goal is to detect and localize the middleboxes and to characterize their action. In order to detect those, you will be invited to code a *simplified version* of `tracebox` in Python using the Scapy[2] library.

2 Requirements

- VirtualBox[3] installed on your machine,
- An ability to code in Python 2.x.

3 Importing a VDI in VirtualBox

The exported virtual machine consists in a VirtualBox-specific container file, called Virtual Disk Image (VDI) available through USB-keys during the lab session. If you are not sure of how to import it, follow closely this guideline:

1. Make sure you have downloaded the .vdi file
2. Start VirtualBox and create a new virtual machine (Linux, Ubuntu)
3. When you're asked for a hard disk image, select *Use existing hard disk* and click on the small icon on the right
4. Select the VDI file from step 1 and press *Next*

[1] `http://wiki.netkit.org/index.php/Main_Page`
[2] `http://www.secdev.org/projects/scapy/doc/`
[3] `https://www.virtualbox.org/wiki/Downloads`

5. After leaving the Virtual Media Manager, you'll be back in your virtual machine wizard. Now you can select your the VDI as existing hard disk and finalize the creation process.

6. Back in the main window, you're now able to start the virtual machine

7. Optional: If you want a shared folder to transfer code of your own on the virtual machine, create a readable folder named `/tmp/vboxshared` and add it to the shared folder list (set auto-mount) in the VirtualBox settings. Its content will be available in the `sf_vboxsfared` folder on the virtual machine desktop.

4 System information

- **Machine**: `ubuntu-VirtualBox`
- **OS**: `Ubuntu 12.10 (quantal) 32bits`
- **Kernel**: `3.5.0-17-generic`
- **User**: `ubuntu`
- **Password**: `i<3traceb0x`

5 Materials

The VM needed for the laboratory will be available through USB-keys during the lab session. Other materials is available at `http://queen.run.montefiore.ulg.ac.be/~edeline/`. Follow the guideline of section 3 to import it into VirtualBox. Once you have done it, you can access the following content:

- 6 netkit network topologies (See section 6)
- a simple traceroute written in Python with Scapy that you can use as a base for a tracebox.

6 Network topologies

There are 6 different network topologies placed in ~/Desktop/labs/lab{1-6}. Each of the topologies contains different middleboxes which will affect differently the downloadings. To launch a topology, change the working directory of a terminal to one of the lab{1-6} folder using `cd`. Once you are inside, you are able to set up/manipulate the lab using the following commands:

- `lstart`: starts a netkit lab (NOTE: **allocate more memory with** `--pass=-M128`)
- `lhalt`: gracefully halts all vms of a lab
- `lcrash`: causes all the vms of a lab to crash (**Use with Caution**)
- `lclean`: removes temporary files from a lab directory

Once you have started a lab, several terminals will pop up (one per machine). **ALL YOUR PROBING COMMANDS MUST BE TYPED FROM THE pc0 TERMINAL !** From the netkit terminal, you can access the Ubuntu VM through `/hosthome/`. From the Ubuntu VM, you can share file with the netkit VM (pc0) through `lab{1-6}/pc0/root/`.

7 Questions

1. Launch the first lab (NOTE: **If terminals quickly shut down, allocate more memory with** `--pass=-M128`). Verify that the number of opened terminals corresponds to the number of PCs in the lab folder. If not, halt the lab and restart it.
2. **ALWAYS in the pc0 terminal window**, run several `ping` and `traceroute` to the destination (always `10.10.10.1`) or to any other hop and find the different paths.
3. Build a tracebox-like tool in Python to detect and localize middleboxes and to characterize their action. To achieve it, we suggest you to use the Scapy Python library. There is a Python/Scapy traceroute on the initial working directory of `pc0` (`/root/`) that you can use as a base for your tool. It also contains useful informations about Scapy. If you prefer, you can use dpkt[4] and twisted-python[5] instead of Scapy.
4. Describe the path from the source host (`20.20.20.2`) to the destination host (`10.10.10.1`), the middleboxes you detected in the previous step by the type of modifications they apply on packets and their location, and any other network component you detect (e.g: load balancer, ...).
5. Repeat previous steps with the following lab.

8 Documentation

- The Python documentation: `https://docs.python.org/2/`
- The Scapy documentation: `http://www.secdev.org/projects/scapy/doc/`
- The dpkt documentation: `http://www.commercialventvac.com/dpkt.html`
- The twisted-Python documentation : `https://twistedmatrix.com/trac/wiki/Documentation`

9 VirtualBox Troubleshooting

If the VM is too slow, try these steps:

1. Shutdown the Ubuntu VM
2. Enable the 3D acceleration in Settings>Display
3. Start the VM

10 Feedback

To obtain feedback about your work, send **a PDF** with your name and answers **AND** your Python simple tracebox at korian.edeline@ulg.ac.be.

References

[1] G. Detal, B. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet. Revealing middlebox interference with tracebox. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 1–8. ACM, 2013.

[4] `http://code.google.com/p/dpkt/`
[5] `https://twistedmatrix.com/trac/`