Snort - Network Intrusion Detection & Prevention System on Ping Flood using Core Emulator

Jessica Arruda Ferreira de Santana
Master student at PUCRS

Why are we talking about this?

- We want to provide **an easy way** to understand some topics of computer network;
- We want to **teach the reader** the benefits of Core Emulator platform and what it is capable of doing;
- We also want to introduce the reader to two center topics: **an attack of DDOS called Ping Flood and the framework Snort.**

What is Core Emulator and why are we using it?

- The Common Open Research Emulator (CORE) is a **tool for emulating networks on one or more machines**. You can connect these emulated networks to live networks. CORE **consists of a GUI for drawing topologies of lightweight virtual machines**, and Python modules for scripting network emulation.
- CORE has been developed by a **Network Technology research group that is part of the Boeing Research and Technology division.** The Naval Research Laboratory is supporting further development of this open source project.

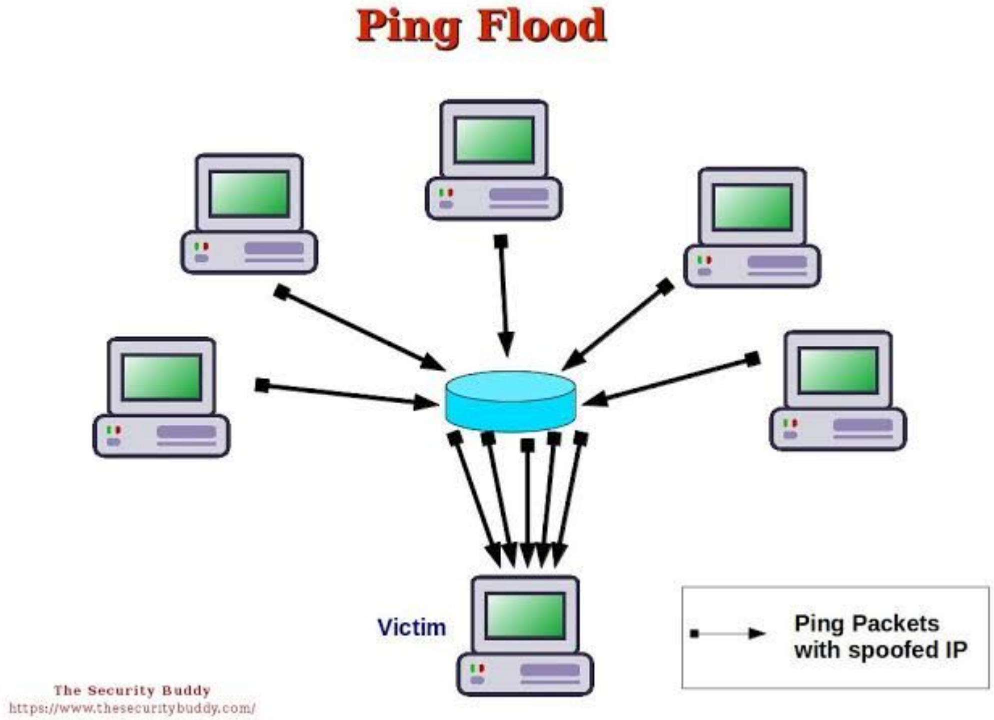
- We choose CORE because it is a **free and open software** that is available on Github;
- It is also **easy to understand** and to start working on it;
- You can have a **nice interaction with the GUI**, where you can build an friendly interface.

Introducing you to Ping Flood!

- Ping Flood i**s a denial-of-service attack** in which the attacker attempts to **overwhelm a targeted device with ICMP echo-request packets**, causing the target to become **inaccessible to normal traffic**.
- When the attack traffic comes from **multiple devices**, the attack becomes a **DDoS** or **distributed denial-of-service attack**.

How does a Ping Flood attack works?

1. A high quantity of attackers send **ICMP packets** to the victim;
2. They **overflow** the victim capability of answering all the packets;
3. The victim gets so busy trying to respond it all, that gets **unable to respond anyone else**.


Ping Flood
Victim
Ping Packets with spoofed IP
The Security Buddy
https://www.thesecuritybuddy.com/

And now we step into Snort! But, what is it?

- Snort is an **open source network intrusion prevention system**, capable of performing **real-time traffic analysis and packet logging on IP networks**. It can perform protocol analysis, content searching/matching, and can be used to **detect a variety of attacks and probes**.

Going a bit deeper...

- Snort can be configured in three main modes: **sniffer, packet logger, and network intrusion detection**.
- In **sniffer mode**, the program will read network packets and display them on the console.
- In **packet logger mode**, the program will log packets to the disk.
- In **intrusion detection mode,** the program will monitor network traffic and analyze it against a rule set defined by the user. The program will then perform a specific action based on what has been identified.
- We will focus on the **intrusion detection mode**.

So... Connecting the dots:

- We introduced you guys to a platform: **Core Emulator.** It is capable of emulate networks and to define your own configurations.
- We also told you about **Ping Flood**, an **DDOS Attack** that mean people use to impossibilitate an victim to perform its communication;
- And **Snort**, which is a friendly software that **warns us whenever some attack is happening** based on a set of rules.

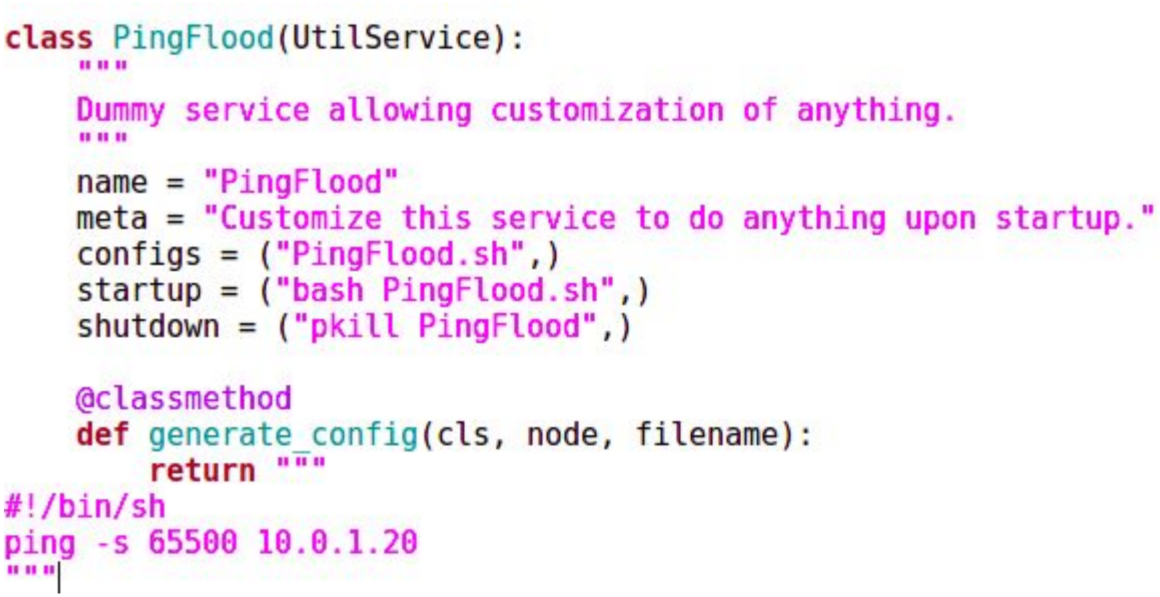
Let’s put it all together then!

- As you could see, all the 3 topics can be related.
- Let’s try to learn 3 different things today:
    - Can Core Emulator really emulate an specific network where we can reproduce an Ping Flood attack and see Snort working?
    - Can we detect an Ping Flood attack using Snort?
    - Is the Ping Flood attack really that harmful?

Time to work (Part 1)

1. Download **Core Emulator;**
2. Download **Snort** using the following commands:
   a. apt-get install libpcap-dev bison flex
   b. apt-get install snort
   c. To verify if it all went okay, run: snort -v
3. Now we are going to create a service called **Ping Flood.**
   a. cd /usr/lib/python3/dist-packages/core/services
   b. sudo gedit utility.py
   c. We are going to type into the file the following text

Time to work

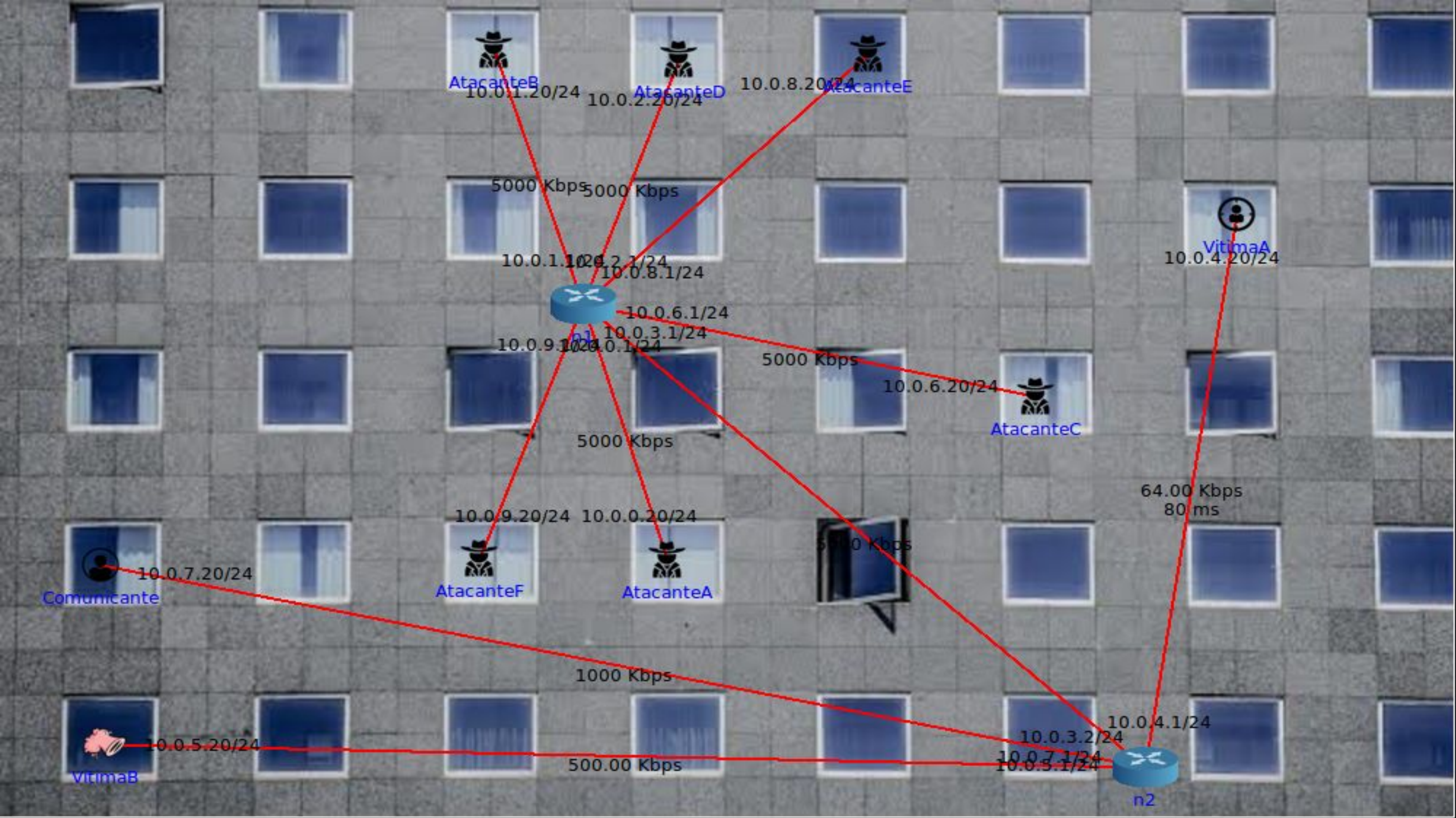


```
class PingFlood(UtilService):
    """
    Dummy service allowing customization of anything.
    """
    name = "PingFlood"
    meta = "Customize this service to do anything upon startup."
    configs = ("PingFlood.sh",)
    startup = ("bash PingFlood.sh",)
    shutdown = ("pkill PingFlood",)

    @classmethod
    def generate_config(cls, node, filename):
        return """
#!/bin/sh
ping -s 65500 10.0.1.20
"""
```

Time to work (Part 2)

1. Start Core Daemon with the following command
   a. sudo service core-daemon start
2. Open Core-Gui with the following command.
   a. sudo Core- Gui
3. Now let’s build a topology that looks like this one in the next image.
   a. For that, all the images used in the project are available on the github repository. The directory is called **image.**
   b. You should save this images and copy them to the path: /usr/share/core/icons/normal
   c. To do that, you can use the following command: sudo cp <name-of-the-file>.gif /usr/share/core/icons/normal


AtacanteB
10.0.1.20/24
AtacanteD
10.0.2.20/24
10.0.8.20/24
AtacanteE
5000 Kbps
5000 Kbps
VitimaA
10.0.4.20/24
10.0.1.1/24
10.0.2.1/24
10.0.8.1/24
10.0.6.1/24
10.0.3.1/24
n1
10.0.9.1/24
10.0.0.1/24
5000 Kbps
10.0.6.20/24
AtacanteC
5000 Kbps
64.00 Kbps
80 ms
10.0.9.20/24
10.0.0.20/24
5000 Kbps
10.0.7.20/24
Comunicante
AtacanteF
AtacanteA
1000 Kbps
10.0.4.1/24
10.0.3.2/24
10.0.5.20/24
10.0.7.1/24
10.0.5.1/24
500.00 Kbps
VitimaB
n2

Time to work (Part 3)

Now it is time to add our services to the attackers (the spy guys).

1. Click twice in the attackers and go to services. In there (if you made **part 1** correctly), you'll find a service called **Ping Flood.**
2. Click on it and then click on the edit button right beside its name.
3. Now we are going to define 3 attackers for victim.
4. For that you'll have to check your victims IP addresses. On my topology they are: 10.0.4.20/24 and 10.0.5.20/24.
5. We are going to make 3 of these attackers to aim on the 10.0.4.20/24 machine and the other ones on the 10.0.5.20/24.
6. After that, we are going to click apply and when we go back to the first page, the edit button is going to be green.

Time to work (Part 4)

We are almost done. Now we need to configure the **bandwidth**.

1. On the topology, click twice on the red line that goes from 10.0.4.1 to 10.0.4.20.
2. Set its bandwidth from **unlimited to 64k**.

Execute!

Well done! Now if we configured everything as it should, we'll see the results now.

- Click on the play button.
- Go to widgets on the **upper menu** and select the **throughput** option to see all the bandwidth going.
- On the beginning the attackers won't arrive to its victims because the routers don't know each other yet.
- While they are making that exchange of packets, we'll go to the **snort computer** and activate it using the following command
  - snort -A console -q -i eth0
- Now if we wait a while, after the routers made their connections, the attackers will **start to overflow the victims** with packets.
- If we try to send a normal ping from the machine 10.0.7.20 to the machine 10.0.4.20, we'll receive a message from the router called **destination unreachable** after a while. (which means that our **ping flood attack works!**)
- And when we open **snort terminal** (click twice on the snort machine), we'll see lots of messages **warning** us that it is receiving lots of packets coming from the attackers.

Conclusion

- The answer for our 3 questions is YES!
  - We can emulate an specific network where we can reproduce an ping flood attack and see snort working.
  - We can detect an Ping Flood attack using Snort.
  - And yes, a Ping Flood attack is really that harmful.
- Besides that implementation, Core Emulator can do much more! With a little bit of time and patience you can emulate numerous networks scenarios.
- It is worth the try.

Thank you!