INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Forensics Cyber Security

MEIC, 2016-2017 – 1st Semester

BitTorrent Traffic Detector

(360BitScoping)

Group 24

78264 Pedro Guerreiro, 78958 Gonçalo Rodrigues, 78328 Pedro Santos

1 Description of functionality

- The tool is **expected to track down the sources of incoming BitTorrent traffic** within a local network, since this content can be illegal.
- To achieve this goal, it should be able to **collect all packets within the local network**.
- To identify what packets are related to BitTorrent traffic the tool will **filter through all the collected packets**.
- It will also **output relevant information about the captured traffic** such as:
  - The downloader's IP address.
  - The name of the torrent being downloaded, if possible.
  - Size of total downloaded data.
  - The date and time of the oldest and newest BitTorrent Packets received.
  - A complete capture output file, if requested.

2 Description of the architecture

The project will be consisted of the following components:

- **Sniffing component (tcpdump)** - TCPDump will be used in order to collect data from the network into a file that will be filtered accordingly.
- **Protocol Filters/parsers** - This is the main functionality of the tool. It should analyze a packet and detect whether or not it is a bittorrent packet.
- **Output generato**r - When a packet is detected as a bittorrent packet, the output generator should extract relevant information from the stream (such as ip addresses or file metainfo) and display it to the user.
- **UI component** - The application will be controlled through the command line. This includes the arguments such as: verbosity, target network, real time, ip range and capture output file.

3 Plan for the implementation

- Create a **TCPDump FIlter** that minimizes the input received while not losing BitTorrent related packets.(**complex pcap filter**).
- Use the previously created filter to **capture from a network filled with "noise"**, and use that data to **run tests and adapt the parsers.**
- **Implement parsers** for the bittorrent protocol, tracker protocol and p2p protocol, in order to filter and identify relevant data. (**Python**)
- Create the **output generator** to manipulate the identified data and output it in order to **output it in a user friendly way.**
- Integrate all the components in a **terminal based ui**.

Notes: In all the topics mentioned above, casual tests will be done whenever is necessary.

It has not yet been considered the case where a virtualized network is segmented (has subnets).

4 Plan for the evaluation

- The evaluation will be done in a controlled and realistic environment, simulating an user normal traffic (Facebook, Youtube, …) and BitTorrent traffic (i.e, the user will be downloading a torrent).

- Wireshark will be used in order to compare the detail of the capture since wireshark is a "general" packet sniffing tool and ours is a BitTorrent only oriented one.

- The capture analysis of the tool must correspond with the information of the Torrent that the user is actually downloading (which is easy to verify, because the torrent is known).