Interception of P2P Traffic in a Campus Network

Merouane MEHDI

Electronics Department, University Blida, Algeria

mehdi_merouane@yahoo.com

Abstract: Nowadays, many universities face bandwidth saturation problem caused by several origins. These include youtube abusively use, online games and especially illegal downloading that makes havoc to Peer-to-Peer protocol e.g. BitTorrent. The latter is often associated with data piracy and copyright violation. This article aims at presenting on one hand the impact of the use of Peer-to-Peer file sharing traffic on campus bandwidth by observation of BitTorrent traffic and on the other a method for limiting the illicit access to this kind of networks. For this purpose, we used a set of open source tools like wireshak sniffer software to capture Bitorrent traffic. Additionally, using the well-known Snort intrusion detection system with a number of adequate and new rules one can reduce bandwidth saturation problem. This solution allowed in our case, a bandwidth saturation reduction of 35%.

Keywords: Intranet, P2P traffic, BitTorrent, µtorrent, Bandwidth, Snort IDS.

1. Introduction

The development of computers and the computer network coupled to the democratization of Internet access facilitates the dissemination of information in a digital form elusive. Each individual connected to the Internet can access a wealth of varied information. Today, almost everything goes through Internet: our message, content that we consult on the Web but also television, live video, etc. This makes data gigabits; the Peer-to-Peer (P2P) activity taking a large part of this network.

It is known that P2P is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session. Unlike the client/server model, in which the client makes a service request and the server fulfils the request, the P2P network model allows each node to function as both a client and server [2].

Is it estimated that for any given network up to 60 to 80% of their traffic is consumed by P2P traffic. People are using P2P application they will consume a huge amount of bandwidth.P2P applications are prone for wide spreading Pirated software. Users might be using pirated software on their computers and auditors will never appreciate that. You can never trust the file you are downloading from a remote user in P2P environment and 90% of the files contain malwares according to Symantec [13]. Thus if your users are using P2P application there is very high rate of virus outbreak in the network and very frequently. In 2015, 10% of malware were propagated via P2P applications. Even the very infamous “W32.Downadup” also propagated and updated itself via P2P applications [2].

BitTorrent is a communication protocol of peer-to-peer file sharing which is used to distribute data and electronic files over the Internet. BitTorrent is one of the most common protocols for transferring large files, such as digital video files containing TV shows or video clips or digital audio files containing songs. Peer-to-peer networks have been estimated to collectively account for approximately 43% to 70% of all Internet traffic. For instance, in November 2014, BitTorrent was responsible for 25% of all Internet traffic. As of February 2015, BitTorrent was responsible for 35% of all worldwide bandwidth [3]. In early 2016, AT&T estimates that BitTorrent represented a quarter of all Internet traffic. BitTorrent and µTorrent software client surpass 150-million user milestone [4].

The impact of using µTorrent client made the same havoc on the bandwidth than an attack of denial of service (DoS), since this is a p2p downloading not detected by the IDS and operates full time. The size of downloading files from P2P network continues to grow, such as video, particularly movies download that upgrades a 720x300 resolution for a 700 MB to a current size of 20 GB or more for a 4K resolution. This is also noted for games. The size can be more than 60 GB for games, this imposes a huge bandwidth occupation. A central campus server can not easily tolerate such amount of information, especially with the growing number of illegal websites.

In this context, the Academic Research Network (ARN) in the campus suffered from this situation. HTTP and FTP downloads were hampered due to p2p streaming over the 100 MB available bandwidth.

Thus, the Snort intrusion detection software was used to overcome Torrent download and subsequently generalised to the entire ARN network. The challenge, however, was in detecting the P2P file sharing programs, tracker site and blocking the activity. Knowing that, at first the in and out traffic for peer to peer file sharing occupy up to 10% (in the upload) and 34% (in the download) of the bandwidth in the campus as shown in Figure 1. During working hours, the number of users among teachers, workers and students exceed 50,000 with 3000 simultaneous connexions.

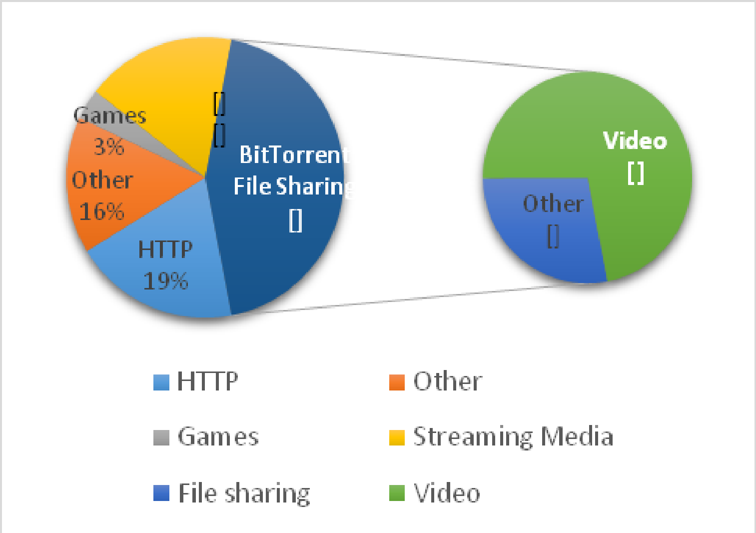

Figure 1. Typical Bandwidth Use in the Campus

In order to improve network services, several steps were taken to reduce p2p traffic:

1. Modeling of the mechanism used by BitTorrent and uTorrent client.
2. Sniff the traffic using appropriate tool to capture p2p packets.
3. Observe the impact on bandwidth.
4. Propose a news rules for detecting traffic P2P.
5. Deal with the challenge of encrypted data.
6. Identify the BitTorrent protocol and particularly the applications.
7. Measure the approach results.

„The ARN network was deployed in the early 90s to provide a technological infrastructure for the benefit of all stakeholders in higher education, scientific research and technological development."

2. BitTorrent protocol

Programmer Bram Cohen, a former University student at Buffalo, designed the protocol in April 2001 and released the first available version on 2 July 2001 and another version later in 2013. BitTorrent clients are available for a variety of computing platforms and operating systems including an official client released by BitTorrent, Inc [5].

The BitTorrent network has a very particular architecture. It is not centralised as was Napster or eDonkey2000, connecting all the peers on one big server. The main risk is that if the server falls this will involve the entire network. It can also lead to legal suite if it considers that this server contains files subject to copyright. That is what happened to Napster, which was closed down in 2002 [5]. Which does not look like a decentralized architecture, that connects peers on multiple servers simultaneously (Emule, Fast Track ...). These servers communicate with each other; however, the system will be more robust.

In fact, the peers are not part of a global network, but they are grouped by file. There is a network around each ".torrent" file. BitTorrent is a set of mini-networks and it is connected only to users with the data to be downloaded and / or shared.

To send or receive files, a person uses a BitTorrent "client" on his Internet-connected computer. A BitTorrent client is a computer program that implements the BitTorrent protocol. Clients include µTorrent, Xunlei, Transmission, qBittorrent, Vuze, Deluge, and BitComet. BitTorrent trackers provide a list of files available for transfer, and allow the client to find peer users known as seeds who may transfer the files. In our case, we noticed µTorrent client, 60% of users in the campus download with this client.

2.1. BitTorrent anatomy:

At BitTorrent architecture, there are two main aspects [6]:

1) Website hosting called very often BitTorrent tracker website whose operation is as follows:

- Start running a tracker.
- Start running an ordinary web server, such as IIS, or have one already.
- Associate the extension .torrent with mimetype application/bittorrent on their web server.
- Generate a metainfo (.torrent) file using the complete file to be served and the URL of the tracker.
- Put the metainfo file on the web server.
- Link to the metainfo (.torrent) file from some other web page.
- Start a downloader, which already has the complete file (the 'origin').

2) The second aspect that interests us is the downloader, where the user downloads p2p file following as follows:

- Install BitTorrent client.
- Surf the web. Enter in the BitTorrent website.
- Click on a link to a .torrent file.
- Select where to save the file locally, or select a partial download to resume.
- Wait for download to complete.
- Tell downloader to exit (it keeps uploading until this happens).

Before making sense of the traces of BitTorrent and µTorrent in captured packets, we will give an overview of the different terminology related to BitTorrent:

Index	An index is a list of .torrent files managed by a website and available for searches. An index website can also be a tracker.
Peer	A peer is one instance of a BitTorrent client running on a computer on the Internet to which other clients connect and transfer data.
Torrent	A torrent can mean either a .torrent metadata file or all files described by it, depending on context. The torrent file contains metadata about all the files that make it downloadable, including their names and sizes and checksums of all pieces in the torrent.
Seed	A seed refers to a machine possessing some part of the data. A peer or downloader becomes a seed when it starts uploading the already downloaded content for other peers to download from.
Leech	The term leech also refers to a peer (or peers) that has a negative effect on the swarm by having a very poor share ratio, downloading much more than uploading.
Swarm	Together, all peers sharing a torrent are called a swarm.
Tracker	A tracker is a server that keeps track of which seeds and peers are in the swarm. Clients report information to the tracker periodically and in exchange, receive information about other clients to which they can connect.

2.2. P2P file sharing program “µTorrent”.

A µTorrent client is program that implements the BitTorrent protocol. Each client is capable of preparing, requesting, and transmitting any type of computer file over a network, using the protocol. A peer is any computer running an instance of a client. To share a file or group of files, a peer first creates a small file called a "torrent". This file contains metadata about the files to be shared and about the tracker, the computer that coordinates the file distribution [8]. Peers that want to download the file must first obtain a torrent file for it and connect to the specified tracker, which tells them from which other peers to download the pieces of the file (Figure 2).

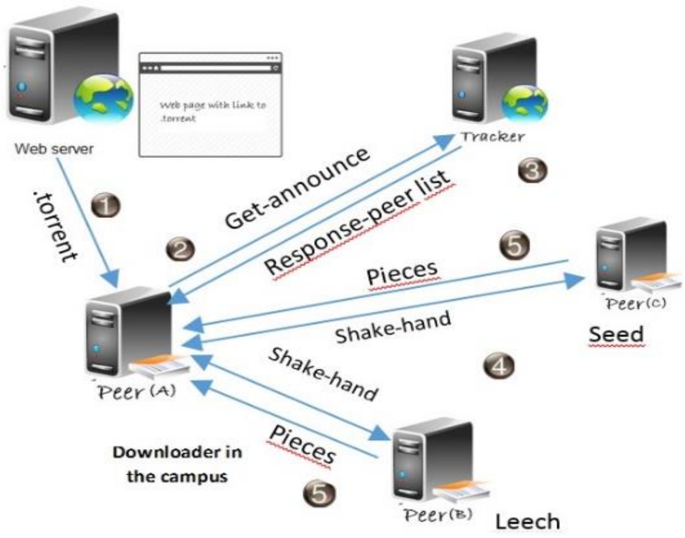

Figure 2. BitTorrent File Sharing Network

3. Materials and methods

We use network monitoring to collect traces of traffic flowing between the University and the rest of the Internet. Campus network connects to its ISPs via a border router, it is used for outbound and inbound traffic. This router is connected to switch. This switch has a monitoring port that is used to send copies of the incoming and outgoing packets to our monitoring host. The entire DMZ passes through the switch, the users of the campus network automatically pass through the proxy server having a local address. The only means of security is the firewall that sits between the router and the switch.

At the monitoring host, the daily magnitude of the consumption of bandwidth for a period of one month with PRTG graph software, and for BitTorrent activity on a campus Network is reported using WireShark sniffer software (Figure 3).

In order to understand more accurately the P2P phenomenon, we installed a BitTorrent client and Utorrent on a PC with a public address connected to the switch. With the help of sniffer Wireshark, we came up in what follows with various clues to detect this p2p download in the intranet of the campus.

A site like RARBG.to that is very popular hosts several types of files to download freely namely the music, books, games, applications and especially high-quality movies. This choice of website is only to give and explain how this kind of torrent tracker works, surely not for free advertising.

One can simply make a ping to determine the IP address. The result was the address 185.37.100.122 for domain “rarbg.to”. As can be easily observed at the capture, the connection to the website is through a link mentioned here https://rarbg.to. With just a click on one of the index mentioned in the website page the user starts downloading through torrent client. The downloaded file has an extension ".torrent".

For the test performed with the client µTorrent and the last release of software which is 3.4.8.A, BitTorrent client normally associates the TCP port number 6881. However, if this port is busy for some reason, the client will instead try successively higher ports (6882, 6883, and so on up to a limit of 6999). With the campus network firewall, we can already start by setting a rule to block these ports. In addition, we can observe the BitTorrent client (µTorrent) port as 60447 (Figure 5), which is being communicated to BitTorrent Server as HTTP Request.

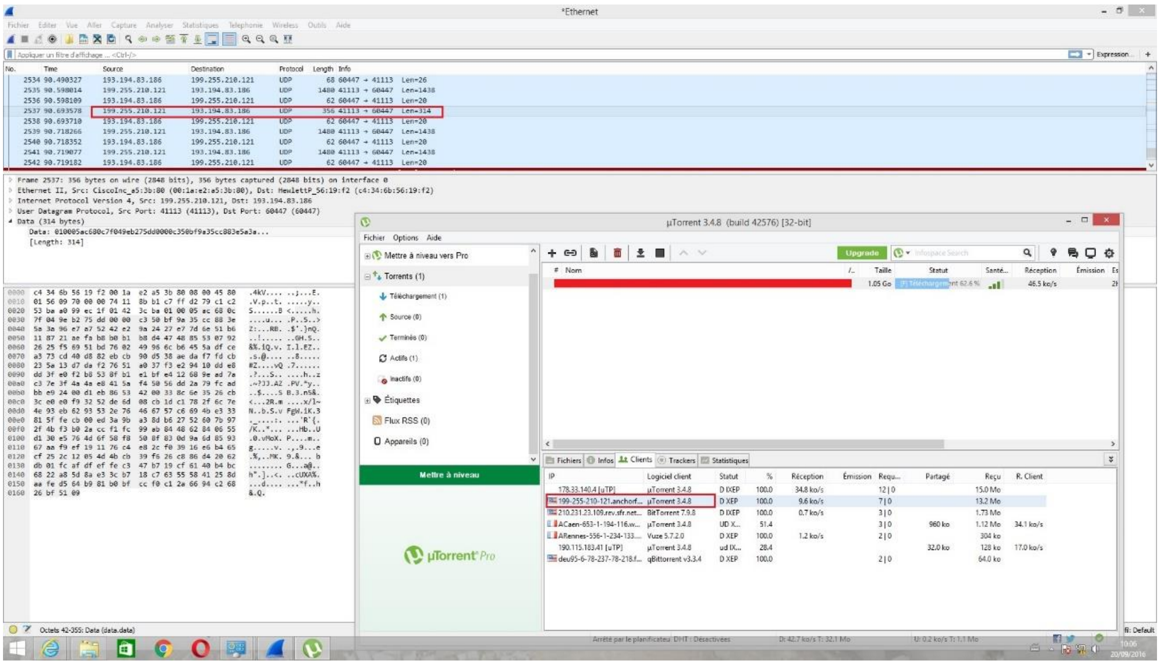

Figure 3. BitTorrent Traces in the Wireshark sniffer software

3.1. Tracker Websites P2P access

For the network administrator a primary operation, is detecting the different torrent tracker website. A recent study dated September 2016, regroups the most popular tracker web p2p, see Table 1.

Table 1. Popular tracker P2P websites

Demonoid.cc	RARBG.to	EZTV.ag	idope.se
Bitport.io	Boxopus	ExtraTorrent.com	Toorgle.com
SeedPeer.au	Isohunt.to	Torrent Funk	Torlock.com

At our campus, wireshark allowed us just to see that the majority of torrent tracker websites are mentioned in this table 1, add to this table a torrent tracker website often consulted by campus users, not known to English speakers as it is a French website "cpasbien.cm". The screenshot below clearly shows the name of torrent tracker website in the sniffer traces (Figure 4).

```
0010  00 46 75 f5 00 00 38 11  e7 25 08 08 08 08 c1 c2   .Fu...8. .%......
0020  53 ba 00 35 c5 f4 00 32  26 76 47 de 81 80 00 01   S..5...2 &vG.....
0030  00 01 00 00 00 00 05 72  61 72 62 67 02 74 6f 00   .......r arbg.to.
0040  00 01 00 01 c0 0c 00 01  00 01 00 00 0b 88 00 04   ........ ........
0050  b9 25 64 7a                                        .%dz
```

Figure 4. Trace of BitTorrent tracker site.

By looking at sniffer traces (Figure 5), we noticed that the torrent "announce" requests over HTTP GET by a torrent tracker is visible clearly; see the screenshot in the Figure 5. To generalize, the information that interests us comes just before the string utorrent "User-Agent". This string precedes every BitTorrent client for http get "announce" or "scrape".

Figure 5. Torrent metafile content "announce"

3.2. Capture metafile ".torrent"

Tests were performed to understand the signatures of BitTorrent protocol and traces of Torrent client. We took the case of a download from a website Tracker by random the file « randonnées en France ». When downloaded, the configuration of µTorrent is by default. Figure 6 below shows clearly the name of the metafile with HTTP Get sniff with wireshark software. Encryption default mode here is not enabled.

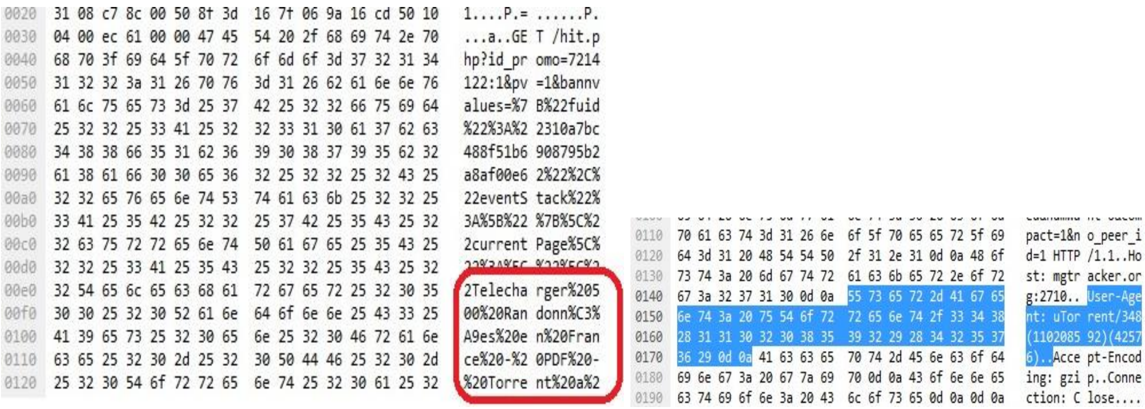

Figure 6. Torrent metafile capture

Furthermore, even the client BitTorrent "uTorrent" can be observed when sniffing packets with the version 3.4.8 build 42576.

Once the user has a connection to peer(s), the first message he sends should be a shake-hand. For the current protocol, it is 'pstrlen' = 19 and 'pstr' = 'BitTorrent protocol' [10]. This string is visible in the sniffer traces (Figure 7).

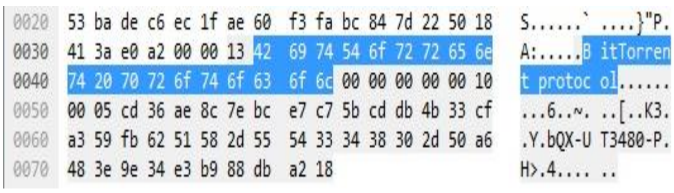

Figure 7. BitTorrent shake-hand capture

So far, the client uTorrent works with a Distributed Hash Table (DHT) disabled, knowing that the DHT is used by BitTorrent clients to find peersvia the BitTorrent protocol. Once the DHT is activated, a ping is used to look after the peers [11]. So there can be detected in tests performed and following DHT ping is represented by this string ‘’d1:ad2:id20‘’ follows with “ping” implemented over UDP protocol. Figure 8 clearly illustrates this appearance. The infohash of the torrent mentioned by “info_hash20” in the sniffer traces associated with a getpeers is represented by “get_peers1”.

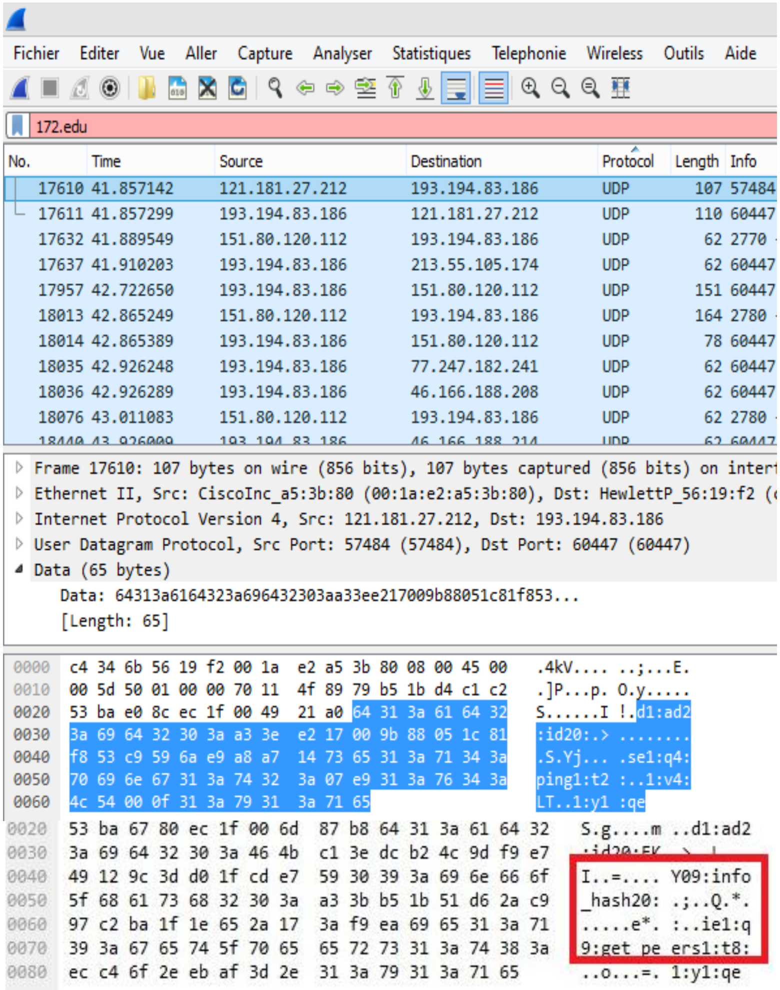

Figure 8. BitTorrent client with DHT enable

3.3. Encrypting file sharing

In most of the detected campus network signatures, the exchange of information in the µTorrent it was not encrypted. However, some aspects of encryption were noticed. Experienced users tried the option of information encryption to avoid detection. Once encrypted, all peer-to-peer exchanges will be encrypted. P2P information exchanged will be only between BitTorrent client that supports encryption. Automatically many sources and peers are reduced and the user has more difficulty to download files. This also reduces the campus available bandwidth.

The current version of wireshark 2.0.5, enables even the detection of the encrypted exchange of utorrent signatures. This is illustrated in Figure 9.

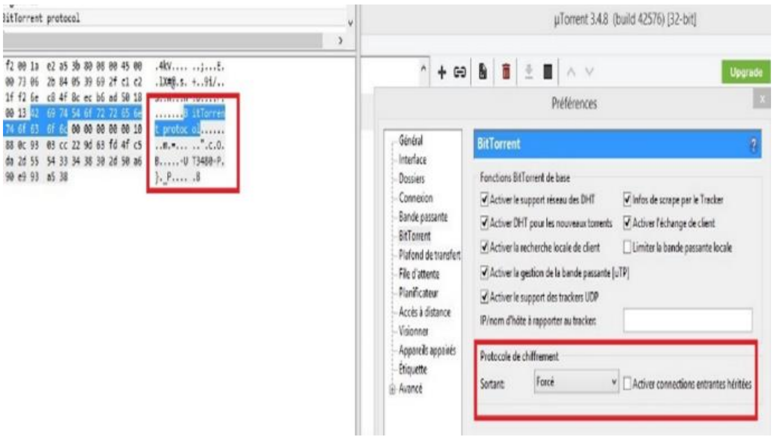

Figure 9. Forced encrypting BitTorrent protocol

Figure 10 shows clearly the signature of «BitTorrent Protocol" in captured packets, followed by the version of µTorrent used in this form "UT3480-P". The traces of the “.torrent” metafile are undetectable because of encryption. With this information, the rules developed in Snort can easily be used.

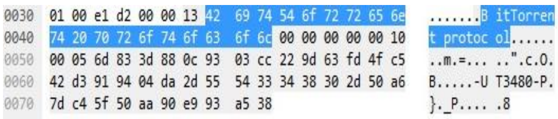

Figure 10. BitTorrent protocol capture

4. Snort rules

Snort intrusion detection system is free and runs on any modern operating system and any old hardware. Snort rules define the patterns and criteria it uses to look for potentially malicious traffic on our network. Without these IDS rules, Snort is just another sniffer like wireshark.

Writing Snort rules is made simple. The language is a follows:

The header of rule that contains:

1. The protocol used for data transmission.
2. The action of the rule.
3. IP source and destination addresses and mask.
4. The source and destination ports.

The rule options (brackets) that contain:

1. The alert;
2. The conditions determining the sending of the alert depending on the inspected package.

The next section explains how to create a customised rule for local use.

The Rule Options section:

Alert is the defined action when a matching signature is detected. The signature in this case is the presence of predefined flags set in the TCP header. Signatures within other rules may be matching payload content, other flags, or binary data (Figure 11).

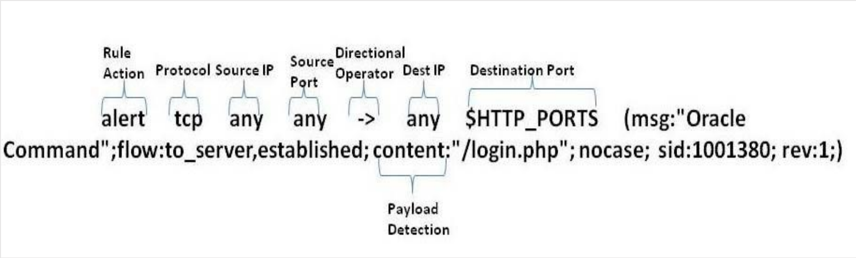

Figure 11. Snort rule description

msg:

The message option explains the type of activity being logged.

content:

The content option is a keyword for defining stings of text or hexadecimal data within the payload. This option is case-sensitive, but can be used with the non-case modifier for case-insensitive matching.

Nocase:

The content modifier nocase deactivates case-sensitivity and looks for matching content.

sid:

This keyword is used to uniquely identify Snort rules.

Classtype:

This keyword is used to categorise a rule as detecting an attack that is part of a more general type of attack class.

The following sample rule is simple and can detect login attempts as root for the telnet protocol (port 23):

```
alert tcp any any ->192.168.1.1/24 23 (msg: " Telnet access attempt for
root ";content: "USER root"; nocase;)
```

A. Detecting BitTorrent in the campus

In the previous section, the sniffer wireshark revealed several clues in the captured packets to the use of BitTorrent. The following Table 2 shows all indices. These indices will enable us to create specific rules in Snort to detect the BitTorrent content.

Table 2. BitTorrent content detection

BitTorrent Message	BitTorrent content “string”
Torrent Tracker website	Rarbg “all tracker web site here” GET /announce /scrape
BitTorrent shake-hand	BitTorrent protocol
Client BitTorrent	Torrent uTorrent User-Agent T3480-P d1:ad2:id20 ping info_hash20 get_peers1

All this information will allow us to develop new Snort rules in order to detect the BitTorrent content. The following rules were implemented:

Detecting Torrent website tracker “case: rarbg.to”:

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg: "Torrent
tracker rarbg"; content:"GET";
content:"rarbg";track by_src;
sid:1000502; rev:2;)
```

This rule is applied to all the torrent tracker websites most popular cited above. The tracker BitTorrent website “cpasbien.cm” is one of the first sites to be detected here at the campus.

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg: "Torrent
tracker cpasbien.cm";
content:"GET";
content:"cpasbien";track by_src;
sid:1000510; rev:2;)
```

Detecting metafile torrent and announce string in the HTTP GET incoming and outcoming:

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any
(msg:”BitTorrent
metafile”;flow:to_server,establi
shed; content:”GET”;
content:”torrent”;
classtype:policy-violation;
sid:1000503; rev:2;)
```

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg:”BitTorrent
http request
out”;flow:to_server,established;
content:”GET”;
content:”/announce”;content:”info_
hash”; classtype:policy-violation;
sid:1000504; rev:1;)
```

BitTorrent client Shake-hand incoming and outcoming:

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg:”BitTorrent
shakehand
in”;flow:to_server,established;
content:”BitTorrent
Protocol”;classtype:policy-
violation; sid:1000505; rev:1;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET
any (msg:”BitTorrent shakehand
out”;flow:from_client,established;
content:”BitTorrent
Protocol”;classtype:policy-
violation; sid:1000515; rev:1;)
```

BitTorrent client with DHT enabled:

```
alert udp $HOME_NET any ->
$EXTERNAL_NET any (msg:”P2P DHT
enable”;content:”d1:ad2:id20”;
content:”ping”;
classtype:policy-violation;
sid:1000506; rev:2;)
```

```
alert udp $HOME_NET any -> $EXTERNAL_NET
any (msg:”P2P DHT get
peers”;content:”d1:ad2:id20”; nocase;
content:”info_hash20”; nocase;
content:”get_peers1 classtype:policy-
violation; sid:1000516; rev:2;)
```

Using the 3.4.8 version of uTorrent is easy to detect users in the campus network with this rule:

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg:"P2P
Utorrent";flow:to_server,establi
shed;content:"User-Agent:
uTorrent";classtype:policy-
violation; sid:1000508; rev:1;)
```

With different BitTorrent client, we change only the name of torrent application.

5. Discussion

A network administrator usually needs a monitoring system that allows a detailed view of transactions within the network. The campus computer network has experienced for some years huge saturations. Studies were carried out to assess the impact of P2P on bandwidth. The first step was modelling BitTorrent fingerprint on the network, the sniffer wireshark has proven to be a useful tool to monitor closely the signature of this P2P network which constituted mainly of Bittorrent client software installed at the user side and also tracker web site offering such metafile torrent download.

The campus network users have a variety of customer p2p softwares such as "qbittorrent, vuze, µTorrent, Transmission, Tixati, Deluge" and many others. The study especially focused on utorrent client since it was wide spread. A number of rules have been developed in order to detect the signatures.

It was found that detecting P2P tracker website is the same as detecting sites that connect to the download file ".torrent". The website "rarbg.to" is a good example, but in reality, there are plenty of other websites that offers the same service. Also, their number is increasing; therefore, the development of rules to detect this kind of website can also be generalized to other websites. At the campus we have developed about 50 rules that regroup popular websites tracker and BitTorrent clients. The proportion of snort rules trigged for BitTorrent traffic in the campus for 3 Days in working hours is illustrated in Figure 12, knowing that all the tests were carried out during a period of 3 months. The number of alerts proves the actual use of P2P in the campus. Visits of torrent website are in continuous grow.

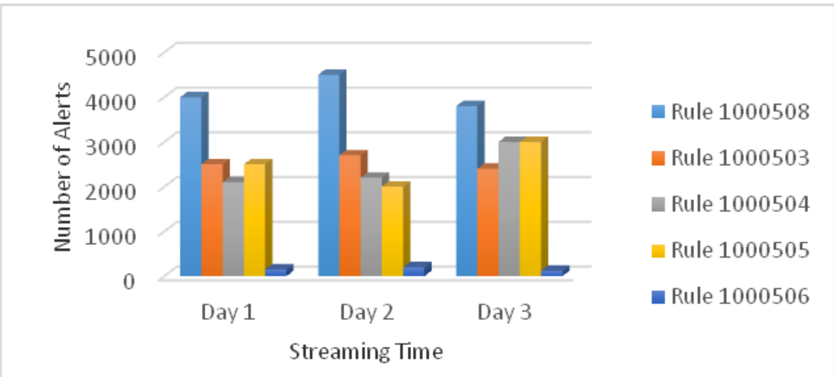

Figure 12. Proportion of Snort rules triggered for Bitorrent traffic in the Campus

The attempt to block these alerts is dealt with in the first place as user management strategy of the campus, by introducing a user directory to the Internet access in the proxy server. A rule is implemented that denies the access to the eliminated sites that use the p2p files. In addition, for BitTorrent client, a warning is issued to those who attempt since each user in assigned in a proper directory. Finally, the firewall can also block ports and unauthorized IP addresses.

With this strategy in place, improvement in browsing the internet locally in the campus was noticed. This is depicted in figure13, which shows bandwidth status before and after rules application. The figure shows data traffic variation for a month before setting up Snort rules and after. It can be easily seen the impact that bandwidth suffers in a daily basis. A clear regression in bandwidth occupation of BitTorrent traffic, can be noticed.

Despite of this effort, the impact of this work remained insufficient since the ARN network includes 134 different institutions all connected to the same backbone. Applying this modest approach to all institutions may lead to be a powerful solution for eliminating this P2P download.

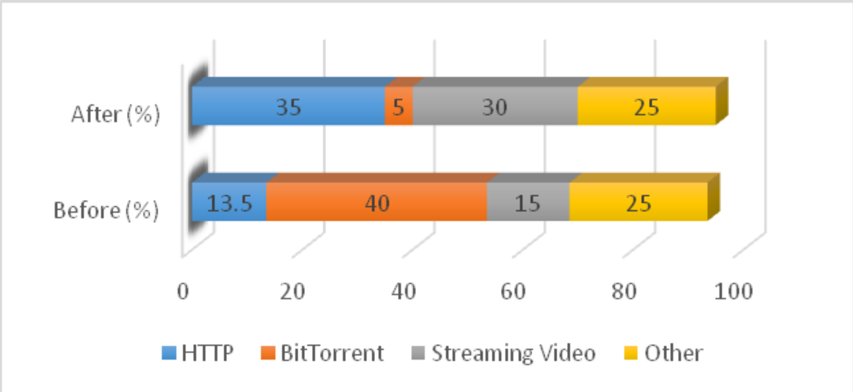

Figure 13. Campus bandwidth categories using

5. Conclusion

The aim of this work was to show the negative impact of the use of P2P such as BitTorrent on the campus network. Observation of the traffic has allowed us one hand to notice the growing number of BitTorrent clients in the range of 60% and on the other to identify and trace the protocol profiles using the wireshark tool.

A strategy was developed based on setting new rules for P2P detection in the Snort IDS and the use of the campus firewall to block undesirable sites and BitTorrent client software. It was possible to detect accurately both in TCP and UDP traffic activity but hard to completely block the access. This is due to the fact that, each download/upload is not from a specific IP address but rather from several different IP addresses. It has also been proven that even with encryption some clues remained. The security strategy implemented, allowed us to improve the consumption of useful bandwidth by 35%. After a month of implementation of this strategy, we observed the occurrence of counter measures by the help of TOR network. This means that strategies need permanently to be updated to ensure healthy network operation.

REFERENCES

1. Cook, T., Conti, G., Raymond, D. (2012). When good Ninjas turn bad: Preventing your students from becoming the threat. *Proceedings of the 16th Colloquium for Information System Security Education*, 2012, 61-67.

2. *** Friend Recommending Peer-To-Peer File Sharing and Synchronization Application, *International Journal of Scientific Research and Engineering Studies (IJSRES)*. Vol. 2 Issue 3, March 2015.

3. *** Application Usage & Threat Report. *Palo Alto Networks*. 2013. Archived from the original on 31 October 2013. Retrieved 7 April 2013.

4. *** AT&T patents system to 'fast-lane' BitTorrent traffic. *Thestack.com*. 8 May 2006. Retrieved 5 March 2015.
5. Pooja Balhara (2016). A Review on Torrent & Torrent Poisoning over Internet. *International Journal of Computer Science & Management Studies*, Vol. 22, Issue 01, 2016.
6. Ying-xu, Hong-guo Yang (2011). https://en.wikipedia.org/wiki/BitTorrent- cite_ref-3 Research on Client Detection of BitTorrent Based on Content. *Communications in Computer and Information Science*. Springer, 473-476, vol. 215, 2011.
7. Ang, Liang, J., Kangasharju (2013). Measuring large-scale distributed systems: Case of Bit Torrent Mainline DHT. *IEEE P2P 2013 Proceedings*, pp. 1–10.
8. *** BitTorrent and µTorrent Software Surpass 150 Million User Milestone. *Bittorrent.com. 9* January 2012. Archived from the original on 26 March 2014.
9. Eric Andre (2004). How to fight P2P in a corporate environment, *SANS Institute 2004, GSEC Practical V1.4b*, Case study in information security. Technical report.
10. Wanner, R. (2009). Detecting Torrents Using Snort. *SANS Institute 2009*, Technical report.
11. Loewenstern, Andrew, Norberg, Arvid (2013). DHT Protocol. *BitTorrent.org*. URL: http://www.bittorrent.org/beps/bep_0005.html, last access in March 22. 2013.
12. Poo Kuan Hoong, Hiroshi Matsuo (2007). A Super-Peer based P2P Live Media Streaming System, *Proceedings of the World Congress on Engineering and Computer Science 2007 WCECS 2007*, October 24-26, 2007, San Francisco, USA.
13. *** What do P2P Applications do? Broadband: fourth report of session, *Symantec report 13*, October 2009.

Merouane MEHDI: I am a teacher and researcher at the Department of Electronics option networks and telecommunications of the University of Blida. and a head of the center of information and communication systems and networks, eLearning and Videoconference. Various articles have been published in the field of computer security.

The main areas of interest for research include: development of computer systems, web developer, education, security network, cybercrime, cyber security, security information, eLearning, smart city.