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 μ Torrent software client surpass 150-million user milestone [4].

The impact of using μ Torrent 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.

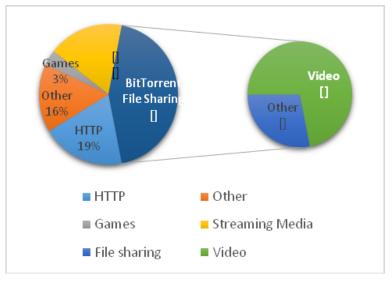


Figure 1. Typical Bandwidth Use in the Campus

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 μ Torrent, 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 μ Torrent 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 μ Torrent in captured packets, we will give an overview of the different terminology related to BitTorrent:

Index	
by a website an index website c	st of .torrent files managed d available for searches. An can also be a tracker.
Peer	
running on a co	enstance of a BitTorrent client mputer on the Internet to ents connect and transfer data.
Torrent	
file or all files of context. The to about all the fil including their checksums of a	hean either a .torrent metadata described by it, depending on rrent file contains metadata es that make it downloadable, names and sizes and ll pieces in the torrent.
Seed	
part of the data becomes a seed	a machine possessing some A peer or downloader when it starts uploading the aded content for other peers om.
Leech	
that has a negat	also refers to a peer (or peers) ive effect on the swarm by oor share ratio, downloading n uploading.
Together, all per called a swarm.	eers sharing a torrent are
Tracker	
which seeds and Clients report in periodically and	erver that keeps track of d peers are in the swarm. nformation to the tracker d in exchange, receive out other clients to which they

2.2. P2P file sharing program "µTorrent".

A μ Torrent 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).

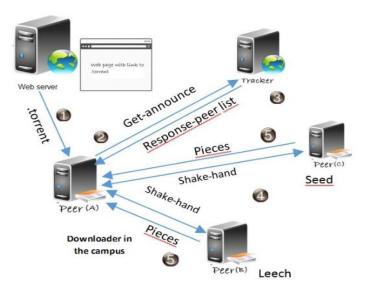


Figure 2. BitTorrent File Sharing Network

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 μ Torrent 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 (μ Torrent) port as 60447 (Figure 5), which is being communicated to BitTorrent Server as HTTP Request.

		"Ethernet		- 4
r Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aid	4			
2 🖲 🧯 🖀 🗙 🖻 🤇 👄 🕾 著 🛓 🛄 🍳 Q, Q, 🗄				
iquer un filtre d'affichage «Ctrl-/>				📰 🔹 Express
Time Source Destination Protocol Length Inf				
	447 → 41113 Len=26 113 → 68447 Len=1438			
	447 + 41113 Len=20			
	113 → 68447 Len=314			
	447 → 41113 Len+20			
	113 → 68447 Len=1438			
	447 → 41113 Len=20 113 → 68447 Len=1438			
	447 + 41113 Len=20			
me 2337: 356 bytes on wire (2048 bits), 356 bytes captured (2048 bits) on i wrmet II, Src: CiscoInc_\$533b100 (00:1a:02:053b100), Dat Heulettp_5619;7 ernet Protocol Version 4, Src: 109-255-1312, Dati 103-134-03-106 m Datagram Protocol, Src Porti 41113 (41113), Dat Port: 60447 (60447) a (314 bytes)		µTorrent 3.4.8 (build 42576) [3;	1601	
Data: 010005ac680c7f049eb275dd0000c350bf9a35cc883e5a3a		µroment 5.4.6 (Duild 42576) [32	-04)	
[Length: 314]	Fichier Options Aide	^ + e9 b a t b ^ ~	Upgrode 🔯 • Infoipace Search	9 9 8 0 0
	⊕ O Mettre à niveau vers Pro		/. Taile Stetut Santu	
	☐ [†] _a Torrents (1)		1.05 Go 11 Teleforment 62.6 %	
C 3 5 6 5 5 19 7 2 00 10 2 2 3 5 0 0 8 0 0 4 5 0 4 0 4 0 4 0 4 0 4 0 5 10 7 2 0 10 10 10 10 10 10 10 10 10 10 10 10 1				
d1 30 e5 76 4d 6f 58 f8 50 8f 83 0d 9a 6d 85 93 .0.vNoX. Pm 67 aa f9 ef 19 11 76 c4 e8 2c f0 39 16 e6 b4 65 gv9e	Appareils (0)	¢		>
cf 25 2c 12 85 4d 4b cb 39 f6 26 c8 86 44 28 62 .X,NK. 9.& b db 01 fc af df ef fe c3 47 b7 19 cf 61 40 b4 bc		v El Fichiers 🕘 Infos 🎎 Clients 🛞 Trackers 🖾 Statistiques		\$
68 22 a8 5d 8a e3 3c b7 18 c7 63 55 58 41 25 8d h".]<,cUXA%.	Mettre à niveau	IP Logiciel client Statut %	Réception Émission Requ Partagé Recu	R. Client
aa fe d5 64 b9 81 b0 bf cc f0 c1 2a 66 94 c2 68d*fh 26 bf 51 09 8.0.		178.33.140.4 [uTP] uToment 3.4.8 D IXEP 100.0	34.8 ko/s 1210 15.0 Mo	
20 01 02 09 8.0.		199-255-210-121.anchorf µToment 3.4.8 D XEP 100.0	9.6 ko/s 710 13.2 Mo	
		210.231.23.109.rev.sfr.net BitTorrent 7.9.8 D IXEP 100.0	0.7 ko/s 3 0 1.73 Mo	
		LACaen-653-1-194-116.w., µToment 3.4.8 UD X., 51.4	3 0 960 ko 1.12 Mo	
		Aflennes-556-1-234-133 Vuze 5.7.2.0 D XEP 100.0	1.2 ko/s 2 0 304 ko	
	UTorrent Pro	190.115.183.41 (uTP) μTorrent 3.4.8 ud IX 28.4	32.0 ko 128 ko	
	Pionent Pio	Beu95-6-78-237-78-218.f., gBittorrent v3.3.4 D XEP 100.0	210 64.0 ko	
Octes 43-000 Data (data data)		Arrête par le planificates DHT : Descrivees Dr.4	L7 ks/s Tr 32.1 Mo UF 0.2 ks/s Tr 1.1 Mo	T 9 0

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 t	racker 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	.Fu8%
0020	53	ba	00	35	c5	f4	00	32	26	76	47	de	81	80	00	01	S52 &vG
0030	00	01	00	00	00	00	05	72	61	72	62	67	02	74	6f	00	<pre></pre>
0040	00	01	00	01	с0	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".

0020	f4	a1	ca	db	1b	39	90	36	22	52	b1	90	c5	f6	50	18		"RP.
0030							-	45			2f					-		T /annou
0040	all shows a second		65				-				61					2.2.2	and the second se	hash=%c
0050			25								62					10000		~%bc%e7%
0060	63		25								62				63	66		%dbK3%cf
0070	25	61	33	59	25	66	62	62			26							QX&peer
0080	69	64	3d	2d	55	54	33	34	38	30	2d	50	25	61	36	25		80-P%a6%
0090	32	30	25	62	35	44	25	61	38	25	62	64	25	64	64	25	20%b5D%a	8%bd%dd%
00a0	1000		25								34							%c4&port
00b0	Зd		30								6f							ploaded=
00c0	30		64								65							aded=135
00d0	33	36	35	36	38	39	26	6c			74					1000		eft=0&co
00e0	72	72	75	70	74	3d	30	26			79					1.000		key=CC38
00f0	42	31	35	45	26	65	76	65			3d					70		nt=stopp
0100			26								3d							nt=0&com
0110			63								70							o peer i
0120	1000		31								2e						and the second second second second	/1.1He
0130			3a								72							ker.flas
0140			Gf								6f					1.11		s.org:69
0150	36		Ød								67							-Agent:
0160	75		6f								34					1000		/348(110
0170	32	30			39						35						the second se	42576)
0180			63								61							ncoding:
0190	20		7a								6e						2 Mar 1997	onnectio
	100																	
01a0	100		20								Ød						n: Close	

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 μ Torrent 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.

0020	31 08 c7 8c 00 50 8t 3d	16 7† 06 9a 16 cd 50 10	1P.=P.		
0030		54 20 2f 68 69 74 2e 70	aGE T /hit.p		
0040		6f 6d 6f 3d 37 32 31 34	hp?id pr omo=7214		
0050		3d 31 26 62 61 6e 6e 76	122:1&pv =1&bannv		
0060	61 6c 75 65 73 3d 25 37	42 25 32 32 66 75 69 64	alues=%7 B%22fuid		
0070	25 32 32 25 33 41 25 32	32 33 31 30 61 37 62 63	%22%3A%2 2310a7bc		
0080	34 38 38 66 35 31 62 36	39 30 38 37 39 35 62 32	488f51b6 908795b2		
0090	61 38 61 66 30 30 65 36	32 25 32 32 25 32 43 25	a8af00e6 2%22%2C%		
00a0	32 32 65 76 65 6e 74 53	74 61 63 6b 25 32 32 25	22eventS tack%22%		
00b0	33 41 25 35 42 25 32 32	25 37 42 25 35 43 25 32	3A%5B%22 %7B%5C%2		countries in outcom
00c0	32 63 75 72 72 65 6e 74	50 61 67 65 25 35 43 25	2current Page%5C%	0 70 61 63 74 3d 31 26 6e 6f 5f 70 65 65 72 5f 69	pact=1&n o_peer_i
00d0	32 32 25 33 41 25 35 43	25 32 32 25 35 43 25 32	22924956 92295692 0120		d=1 HTTP /1.1Ho
00e0			2Telecha rger%205	0 73 74 3a 20 6d 67 74 72 61 63 6b 65 72 2e 6f 72	st: mgtr acker.or
			U UIT		g:2710 User-Age
00f0	30 30 25 32 30 52 61 6e	64 6f 6e 6e 25 43 33 25	00%20Ran donn%C3% 0150	0 6e 74 3a 20 75 54 6f 72 72 65 6e 74 2f 33 34 38	nt: uTor rent/348
0100	41 39 65 73 25 32 30 65	6e 25 32 30 46 72 61 6e	A9es%20e n%20Fran 0160	0 28 31 31 30 32 30 38 35 39 32 29 28 34 32 35 37	(1102085 92)(4257
0110	63 65 25 32 30 2d 25 32	30 50 44 46 25 32 30 2d	ce%20-%2 0PDF%20- 0170	0 36 29 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64	 Acce pt-Encod
0120	25 32 30 54 6f 72 72 65	6e 74 25 32 30 61 25 32	\$20Torre nt%20a%2	0 69 6e 67 3a 20 67 7a 69 70 0d 0a 43 6f 6e 6e 65	ing: gzi pConne
			0190	0 63 74 69 6f 6e 3a 20 43 6c 6f 73 65 0d 0a 0d 0a	ction: C lose

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).

																	S`	}"P.
0030	41	3a	e0	a2	00	00	13	42	69	74	54	6f	72	72	65	6e	A:B	itTorren
0040	74		70	72				63		60	00	00	00	00	00	10	t protoc	ol
0050	00	05	cd	36	ae	8c	7e	bc	e7	c7	5b	cd	db	4b	33	cf	6~.	[K3.
0060	a3	59	fb	62	51	58	2d	55	54	33	34	38	30	2d	50	a6	.Y.bQX-U	T3480-P.
0070	48	3e	9e	34	e3	b9	88	db	a2	18							H>.4	

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".

ichier	r Eo	diter	Vi	ue	Alle	er	Cap	ture	An	alyser	Stat	istique	is Te	lephoi	nie W	lireless	Outils	Aide
(🔳	6	۲		01		(9	÷ •	ء 🖻	Ŷ	Ł		Ð,	ର୍ ବ୍			
172	2.edu																	
0.		Time				So	urce				D	estinat	ion			Protocol	Length	Info
17	610	41.	8571	142		12	1.1	81.2	27.2	12	1	93.19	4.83	.186		UDP	107	574
- 17	611	41.	8572	299		19	3.1	94.8	33.1	86	1	21.18	31.27	212		UDP	110	604
17	632	41.	8895	549		15	1.8	0.12	20.1	12	1	93.19	4.83	186		UDP	62	277
17	637	41.	9102	203		19	3.1	94.8	33.1	86	2	13.55	.105	.174		UDP	62	604
17	957	42.	7226	550		19	3.1	94.8	33.1	86	1	51.80	.120	.112		UDP	151	604
18	013	42.	8652	249		15	1.8	0.12	20.1	12	1	93.19	4.83	186		UDP	164	278
18	014	42.	8653	389		19	3.1	94.8	33.1	86	1	51.80	.120	.112		UDP	78	604
18	035	42.	9262	248		19	3.1	94.8	33.1	86	7	7.247	.182	241		UDP	62	604
18	036	42.	9262	289		19	3.1	94.8	33.1	86	4	6.166	.188	208		UDP	62	604
18	076	43.	0110	983		15	1.8	0.12	20.1	12	1	93.19	4.83	186		UDP	62	278
18	110	43	0260	200		10	3 1	<u>0/ 9</u>	22 1	86	1	6 166	188	21/			67	60/
Eth Int Use	herne terne er Da	et i et i ataj	II, Prot gram	Sno occ Pr	:: C 01 V	isc ers	oIn ion	:_a5 4,	:3b Snc	80 (121	00:1a .181.	:e2:a 27.21	5:3b: 2, Ds	80), t: 19	Dst: 93.194			
Eth Int Use Dat	herne terne er Da ta (0 Data	et : et ata 65 a: (II, Prot gram byte 5431	Sno coco n Pr es) .3a6	: C ol V roto	isc ers col	oInd ion , Sr	:_a5 4, rc P	Src Src	80 (121 574	00:1a .181.	:e2:a 27.21 7484)	5:3b: 2, Ds , Dst	80), t: 19 : Port	Dst: 93.194 t: 604	Hewlet	tP_56:1 6	
> Eth > Int > Use • Dat	herne terne er Da ta (0	et : et ata 65 a: (II, Prot gram byte 5431	Sno coco n Pr es) .3a6	: C ol V roto	isc ers col	oInd ion , Sr	:_a5 4, rc P	Src Src	80 (121 574	00:1a .181. 84 (5	:e2:a 27.21 7484)	5:3b: 2, Ds , Dst	80), t: 19 : Port	Dst: 93.194 t: 604	Hewlet	tP_56:1 6	
> Eth > Int > Use • Dat	herne terne ta (0 Data [Ler c4	et i atap 65 i a: 0 ngti 34	II, Prot gram byte 5431 n: 6 6b	Sno cocc Pr (s) (3a6 (5) (5)	:: C ol V oto 164	isc ers col 323	oIn ion , Sr a690	c_a5 4, rc P 5432 La	src Src 303a	: 80 (: 121 : 574 : 333e	00:1a .181. 84 (5 221700 80 0	:e2:a 27.21 7484) 99b88	5:3b: 2, Ds , Dst 051c8 45 00	80), it: 19 : Port 1f85:	Dst: 93.194 t: 604 3 4kV	Hewlet .83.18 47 (60	tP_56:1 6 447) E.	
> Eth > Int > Use • Dat	herne terne ta (e Data [Ler c4 00	et i ata 65 i a: (ngti 34 5d	II, Prot gram byte 5431 n: 6 6b 50	Sno coco Pr (s) (3a6 (5) (5) (5) (5)	:: C ol V ooto 164 19 00	isc ers col 323 f2 00	oInd ion , Sr a690 00 1 70 1	2_a5 4, FC P 5432 La	:3b Src Port 303a e2 a 4f 8	80 (121 574 a33e 5 3b	00:1a .181. 84 (5 ≥21700 80 00 b5 11	:e2:a 27.21 7484) 99b88 3 00 5 04	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2	80), it: 19 : Port 1f853	Dst: 93.194 t: 604 3 4kV]Pp	Hewlet .83.18 47 (60 ;. ;.	tP_56:1 6 447) E.	
Eth Int Use Dat	herne terne ta (e Data [Ler c4 00 53	et i ata 65 i a: (ngti 34 5d ba	II, Prot gram 5431 n: 6 50 6b 50 e0	Src cocc Pr s) 3a6 5] 56 01 8c	:: C ol V oto 164 19 00 ec	isc col 323 f2 00 1f	oInd ion , Sr a690 00 1 70 1 00 4	(_a5 4, rc P 6432	:3b Src Port 303a e2 a 4f 8 21 a	:80 (: 121 : 574 :a33e :5 3b :5 3b :9 79 :0 64	00:1a .181. 84 (5 21700 80 00 55 10 31 3	:e2:a 27.21 7484) 99b88 99b88 3 00 9 d4 1 61	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32	80), it: 19 : Port 1f85: 	Dst: 93.194 t: 604 3 4kV]Pp	Hewlet .83.18 .47 (60 ;. . 0.y. I !.d1	tP_56:1 6 447) E.	
000 010 030	herne terne ta (0 Data [Ler c4 00 53 3a	et : et atap 65 65 ngth 34 5d ba 69	II, Prot gram 5431 n: 6 50 60 64	Snc cocc Pr (s) 3a6 5] 56 01 8c 32	:: C bl V roto 1164 19 00 ec 30	isc ers col 323 f2 00 1f 3a	oInd ion , Sr a696 00 1 70 1 00 4 a3 3	C_a5 4, FC P 5432 La La 19 Se	e2 a e2 a e2 a e2 a e2 a	80 (121:574 a33e 5 3b 9 79 0 64 7 00	00:1a .181. 84 (5 221700 80 00 b5 11 31 3 9b 8	:e2:a 27.21 7484) 99b88 99b88 9 00 9 d4 9 61 8 05	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32 1c 81	80), t: 19 1f853	Dst: 93.194 t: 604 3 4kV]Pp id20:.	Hewlet 4.83.18 447 (60 ;. 0. 0.y. I !.d1 >	tP_56:1 6 447) E. 	
 Eth Int Use Dat 0000 0010 0020 0030 0040 	herne terne ta ((Data [Ler c4 00 53 3a f8	et : et ata 65 a: (ngth 34 5d ba 69 53	II, Prot gram 5431 1: 6 50 6b 50 e0 64 c9	Snc cocc Pr (s) 3a6 5] 56 01 8c 32 59	:: C bl V roto 164 19 00 ec 30 6a	isc ers col 323 f2 00 1f 3a e9	oInd ion , Sr a690 00 1 70 1 00 4 a3 3 a8 a	c_a5 4, cc P 5432 La 11 49 8e 67	e2 a e2 a e2 a e2 a e2 a e2 a e2 a e2 a	80 (121 574 333e 5 3b 9 79 0 64 7 00 2 65	00:1a .181. 84 (5 221700 80 00 b5 11 31 30 9b 80 31 3	:e2:a 27.21 7484) 99b88 3 00 0 d4 61 3 05 a 71	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32 1c 81 34 3a	80), st: 19 : Port 1f85: S	Dst: 93.194 t: 604 3 4kV]Pp id20:. S.Yj.	Hewlet 4.83.18 447 (60 ;. 0. 0.y. I !.d1 > sel	tP_56:1 6 447) E. .ad2 .q4:	
 Eth Int Use Dat 0000 010 020 030 040 050 	herne terne ta ((Data [Ler c4 00 53 3a f8 70	et : et ata; 65 a: (mgt) 54 54 53 69 53 69	II, Prot gram byte 5431 1: 6 54 6b 50 e0 64 c9 64 c9 6e	Src (000 Pr (1) (5) (5) (5) (5) (5) (5) (6) (7) (7) (7) (7) (7) (7) (7) (7) (7) (7	:: C bl V ooto 164 19 00 ec 30 6a 31	isc ers col 323 f2 00 f2 00 1f 3a e9 3a	oInd ion , Sr a690 00 1 70 1 00 4 a3 3 a8 a 74 3	2_a5 4, 5432 5432 11 19 8e 57 32	e2 a e2 a e2 a e2 a e2 a e2 a e2 a e2 a	80 (121 574 333e 5 3b 9 79 0 64 7 00 3 65 7 e9	00:1a .181. 84 (5 221700 80 00 b5 11 31 30 9b 80 31 3	:e2:a 27.21 7484) 99b88 3 00 0 d4 61 3 05 a 71	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32 1c 81	80), st: 19 : Port 1f85: S	Dst: 93.194 t: 604 3 4kV]Pp id20:. S.Yj ing1:t	Hewlet .83.18 .47 (60 ; .0.y. I !.d1 se1 .21	tP_56:1 6 447) E. .ad2 .q4:	
> Eth > Int > Use • Dat 0000 0010 0020 0030 0040 0050 0050	herna terna ta ((Data [Ler c4 00 53 3a f8 70 4c	et : et ataj 65 a: (54 54 69 53 69 54	II, Prot gram byte 5431 1: 6 54 6b 50 e0 64 c9 64 c9 6e	Src cocc Pr 3a6 5] 56 01 8c 32 59 67 0f	:: C bl V ooto 1164 19 00 ec 30 6a 31 31	isc ers col 323 f2 00 f2 00 1f 3a 29 3a 3a	oInd ion , Sr a690 00 1 70 1 00 4 a3 3 a8 a 74 3 79 3	c_a5 4, fc P 5432 La 11 49 8e 47 32 31	<pre>src: Src: Port: 303a e2 a 4f & 21 a e2 1 14 7 3a 0 3a 7</pre>	80 (121 574 333e 53b 979 0 64 700 3 65 7 9 7 9 1 65	00:1a .181. 84 (5 221700 80 00 b5 11 31 3 9b 8 31 3 31 3	e2:a 27.21 7484) 99b88 99b88 8 00 0 d4 61 8 05 a 71 a 76	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32 1c 81 34 3a 34 3a	80), st: 19 : Port 1f85: S	Dst: 93.194 t: 604 3 4kV]Pp id20:. 5.Yj ing1:t T1:y	Hewlet: .83.18 .47 (60 ;. .0.y. I !.d1 > .se1 :2 :1 1 :qe	tP_56:1 6 447) E. .:ad2 .:q4: :v4:	9:f2
 Eth Int Use Dat 0000 010 020 030 040 040 050 060 020 	herno terno ta ((Data [Ler c4 00 53 3a f8 70 4c 53	et : et I atai 65 I a: (ba 54 53 69 54 ba	II, Prot gram 5431 55431 1: 6 50 60 64 c9 64 c9 60 00	Src cocc Pr s) 3a6 5] 56 01 8c 32 59 67 0f 80	:: C ol V roto 1164 19 00 ec 30 6a 31 31 31 ec	isc ers col 323 f2 00 1f 3a e9 3a 3a 3a 1f	oInd ion , Sr a690 170 1 000 4 a3 2 a8 a 274 2 79 3	c_a5 4, rc P 5432 La 11 19 8e 37 32 31 0 6d	<pre>src: Src: 20rt: 303a 4f & 21 a 4f & 21 a 3a 7 3a 7 1 8</pre>	80 (121 574 333e 5 3b 9 79 0 64 7 00 7 65 7 65 7 b8	00:1a .181. 84 (5 221700 80 00 b5 11 31 30 31 30 31 30 31 30 64 3	:e2:a 27.21 7484) 99b88 3 00 0 d4 a 61 3 05 a 71 a 76 1 3a	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32 1c 81 34 3a 34 3a 61 6	80), it: 19 : Port 1f85: S S S S S S S S S S S S S S S S S S S	Dst: 93.194 t: 604 3 4kV]Pp id20: 5.Yj ing1:t T1:y 2. S.	Hewlet: .83.18 .47 (60 ;. .0.y. I !.d1 > .se1 :2 :1 1 :qe	tP_56:1 6 447) :E. : :q4: : : nd1:	9:f2
 Eth Int Use Dat Dat 0000 0010 0020 0030 0040 0050 0060 0050 0050	herna terna er Data Data ((C4 00 53 3a 70 4c 53 3a	et : et ata 65 a: (hgt) 54 53 69 54 69 54 69	II, Prot gram byte 54311 1: 6 50 e0 64 c9 64 c9 64 67 67 64	Src (0 Pr (1 Pr (2 S) (3 a 6 (5) (5) (5) (5) (6) (6) (6) (6) (6) (6) (6) (6	:: C ol V roto 1164 19 00 ec 30 6a 31 31 ec 30	isc ers col 323 f2 00 1f 3a e9 3a 3a 1f 3a 1f	oInd ion , Sr a690 200 1 70 1 00 4 a3 3 a8 a 70 1 79 3 5 00 1 46	c_a5 4, cc P 5432 5432 1a 11 19 8e 37 32 31 0 6d i 4b	::3b: Src: Port: 303a 4f & 21 a 21 a 21 a 3a 0 3a 7 1 8 0 5 c	80 (121 574 1538 1538 1979 1064 1700 165 165 13e	00:1a .181. 84 (5 221700 80 00 b5 11 31 3 9b 8 31 3 31 3	:e2:a 27.21 7484) 99b88 3 00 9 d4 a 61 3 05 a 71 a 76 1 3a 2 4c	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 64 32 1c 81 34 3a 34 3a 61 6 9d f	80), t: 11 1f853 1f853 S S S S S S S S S S S S S S S S S S S	Dst: 93.194 t: 604 3 4kV]Pp id20:. s.Yj ing1:t T1:y : S.	Hewlet 4.83.18 447 (60 ;. .0.y. I !.d1 se1 2 :1 1 :qe gfi d20.EL	tP_56:1" 6 447) E. :q4: :v4: nd1:	ə:f2
 Eth Int Use Dat Dat 0000 0010 0020 0020 0040 0050 0040 0050 0060 0020 0060 0020 0060 0020 0040 0050 0040	herne tern Data ((Data (Ler c4 00 53 3a 4c 53 3a 49	et : et atap 65 a: (hgt) 34 54 69 53 69 54 69 54 69 54 69 54 12	II, Prot gram 5431 550 60 64 69 64 69 66 00 67	Src (0 Co (1 Pr (s)) (3a6 (5) (5) (5) (5) (5) (6) (6) (6) (7) (6) (7) (6) (7) (6) (7) (7) (7) (7) (7) (7) (7) (7) (7) (7	:: C ol V roto 164 19 00 ec 30 6a 31 31 ec 30 6a 31 0 6a 31 0 00 6a 31 0 00 6a 31 0 00 6a 31 00 6a 31 00 00 00 00 00 00 00 00 00 00 00 00 00	isc ers col 323 f2 00 1f 3a 3a 3a 1f 3a 3a 1f	oInd ion , Sr a690 200 1 70 1 000 4 a3 2 a8 a 70 2 5 00 a 46 5 cd	c_a5 4, cc P 5432 11 59 64 7 32 31 9 6 4 5 4 b 6 4 5 4 5 7 5 2 31 9 6 6 4 5 4 5 7 5 2 31 9 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	<pre>i:3b: Src: vort: 303a 21 a 21 a 21 a 21 a 21 a 21 a 21 a 21</pre>	80 (121 574 153 153 153 165 165 165 165 165 165 165 165	00:1a .181. 84 (5 221700 80 00 55 10 31 30 31 30 31 30 64 3 dc b	:e2:a 27.21 7484) 99b888 3 00 9 d4 a 61 3 05 a 71 a 76 1 3a 2 4c 1 3a	5:3b: 2, Ds 051c8 051c8 45 00 c1 c2 64 32 1c 81 34 3a 34 3a 61 6 9d f 6e 6	80), tt: 11 1f85: 1f85: 5 5 4 9 e7 5 6 6 ff	Dst: 93.194 t: 604 3 4kV]Pp id20:. 5.Yj ing1:t T1:y : S. : I.	Hewlet .83.18 .47 (60 ;. .0.y. I !.d1 se1 se1 e gf 	tP_56:1 6 447) :E. : :q4: : : nd1:	ə:f2 ad2 nfo
> Eth > Int > Use • Dat	hernd er Di ta ((Data [Ler c4 00 53 3a 68 70 4c 53 3a 3a 49 5f	et : et atap 65 a: (mgth 54 54 54 54 54 54 54 54 54 54 54 54 54	II, Prot gram 5431 65 60 64 60 64 60 67 64 90	Src (0 C (1 Pr (5) (3) (5) (5) (5) (5) (5) (6) (7) (6) (7) (7) (7) (7) (7) (7) (7) (7) (7) (7	:: C)1 V roto 164 19 00 ec 30 6a 31 91 90 6a 31 91 90 6a 31 91 90 6a 31 91 90 6a 31 91 90 6a 31 91 90 6a 31 91 90 6a 31 91 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 31 90 6a 80 80 80 80 80 80 80 80 80 80	isc ers col 323 f2 00 1f 3a e9 3a 3a 1f 3a 3a 1f 3a 1f 3a 3a	oInd ion , Sr a690 00 1 70 1 00 4 a3 3 a8 a a8 a a8 a a8 a a8 a a8 a a8 a	c_a5 4, cc P 5432 1a 19 36 37 32 31 9 6d 5 4b 1 e7 9 3a	<pre>::3b: Src: Port: 303a 21 a 4f & 21 a 4f & 3a 0 3a 7 1 8 0 C 5 1 a</pre>	80 (1211 574 1336 153b 153b 165 165 165 13e 930 33b	200:1a .181.: 84 (5 221700 80 00 55 11 31 3; 95 8; 31 3; 64 3 dc b 39 3	227.21 27.21 27484) 99b88 3 00 9 d4 3 05 3 07 1 3a 2 4c 1 3a 2 4c 1 3a 2 4c 5 1	5:3b: 2, Ds , Dst 051c8 45 00 c1 c2 c64 32 1c 81 34 3a 34 3a 61 6 9d f 6e 6 d6 2	80), it: 19 11f85: 11f85: 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 7 2 8 2 9 9 7 5 6 6 6 1 2 8 2 9 9 7 7 6 6 6 1 1 2 8 2 9 9 7 7 9 8 7 9 8 7 9 7 9 7 9 7 9 7 9 7	Dst: 93.194 t: 604 3 4kV]Pp id20:. s.Yj ing1:t T1:y ! S.	Hewlet 4.83.18 47 (60 ; 0.0.y. I !.d1 2 :1 1 :qe 8 420.EL nash20:	tP_56:1 6 447) E. :ad2 : :v4: md1: (ad2 .*.
Eth That Use Dat Dat Dat Dat Dat Dat Dat Dat Dat Dat	herna er Di ta ((Data [Ler c4 00 53 3a 68 70 4c 53 3a 49 5f 97	et : et ata 65 a: (ngth 53 69 54 ba 69 54 ba 69 54 ba 69 54 c2	II, Prot gram byte 5431 1: 6 50 60 64 62 60 64 62 64 90 67 64 90 61	Src (0 C C C C C C C C C C C C C C C C C C C	:: C l V roto 164 19 00 ec 30 6a 31 31 ec 30 6a 31 31 ec 30 6a 31 31 ec 30 6a 31 10 10 10 10 10 10 10 10 10 1	isc ers col 323 323 f2 00 1f 3a 29 3a 3a 1f 3a 3a 1f 3a 3a 1f 32 55	oInd ion , Sr a690 00 1 70 1 00 4 a3 2 a8 a a8 a a8 a a8 a a8 a a8 a a8 a a8	a5 4, cc P 5432 1a 11 9 8 6 31 9 6 6 1 e7 9 3 1 7 32 31 0 6 4 9 1 e7 9 3 1 7 32 31 0 6 4 1 7 32 31 0 7 31 1 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	<pre>i: 3b: Src: ort: 303a 21 a 21 a 21 a 21 a 21 a 3a 0 3a 7 1 8 0 c 7 5 i a a 3 3</pre>	80 (121 121 574 133 15 36 17 16 1 36 17 16 1 37 1 36 1 36 1 36 1 36 1 36 1 36 1 1 1 1 1 1 1 1 1 1 1 1 1	20:1a .181 84 (5 221700 80 00 55 11 31 3 31 3 31 3 31 3 464 3 4c b 39 3 55 1	227.21 27.21 27484) 99b88 90b88 900 900 9000 90	5:3b: 2, Ds 2, Ds 051c8 051c8 45 00 c1 c2 64 32 1c 81 34 3a 34 3a 61 6 9d f 6e 6 d6 2 31 3	80), t: 19 11f85: 5 5 5 6 4 32 5 9 6 6 6 6 6 6 6 7 1 6 6 6 7 1	Dst: 193.194 t: 604 3]Pp id20: . S.Yj ing1:t T1:y S.	Hewlet 433.18 447 (60 	tP_56:1' 6 447) :E. ::q4: :v4: nd1: 	ad2 .*. 1:q

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 μ Torrent 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.

litTorrent protocol v		µTorrent 3.4.8 (build 42576) (32-bit)	I
	1 + CP & T	É ■ A × Préférences	Upgrade
周年初 80 cc 22 56 65 代 4f c5	Interface Dosies Fonctions Comerion Octove Sande pasante Sande pasante Safleuret Plufond de suerden Plufond de suerden Plufond de Active Plufonder Active Active Plufonder Active Plufonder Active Acti	nt Bill Greet de base exsport réseau de DHT	e client ssante locale

Figure 9. Forced encrypting BitTorrent protocol

Figure 10 shows clearly the signature of "BitTorrent Protocol" in captured packets, followed by the version of μ Torrent 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.

0030	01	00	e1	d2	00	00	13	42	69	74	54	6f	72	72	65	6e	B	itTorren
0040	74	20	70		6f	74	6f			6c	00	00	00	00	00	10	t protoc	ol
0050	00	05	6d	83	3d	88	0c	93	03	сс	22	9d	63	fd	4f	c5	m.=	".c.0.
0060	42	d3	91	94	04	da	2d	55	54	33	34	38	30	2d	50	a6	BU	T3480-P.
0070	7d	c4	5f	50	aa	90	e9	93	a5	38							}P	.8

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).

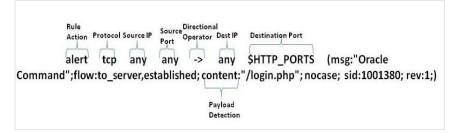


Figure 11. Snort rule description

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.

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

Table 2. BitTorrent content detection

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;)
```

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 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;)
```

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:policyviolation; 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, μ Torrent, 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.

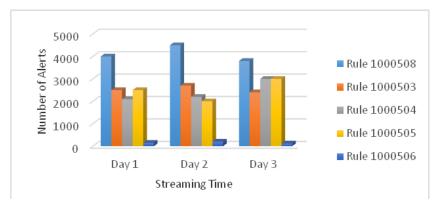


Figure 12. Proportion of Snort rules triggered for Bitorrent traffic in the Campus

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.

http://www.rria.ici.ro

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.

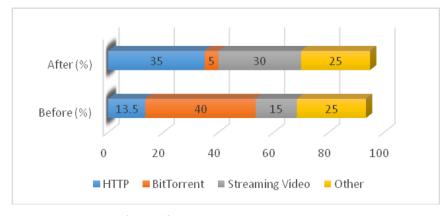


Figure 13. Campus bandwidth categories using

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.
- *** 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.