Análise de metadados no tráfego do protocolo BitTorrent

Tiago da Silveira Pasa[1], **Carlos Vinícius Rasch Alves**[1], **Eduardo Maroñas Monks**[1]

[1]Curso Superior de Redes de Computadores
Faculdade de Tecnologia SENAC Pelotas (FATEC)

tiagopasa@hotmail.com, cvalves@senacrs.edu.br, emmonks@gmail.com

Abstract. *This paper presents techniques to detect BitTorrent protocol and its metadata. Tests comparing DPI techniques and signature-based detection, using open source and commercial tools, were performed. The packages were taken from various torrent sites, with and without encryption enabled. The chosen tools allowed the differentiation between legal and illegal traffic.*
Keywords: BitTorrent, deep packet inspection, signatures, IDS, tests.

Resumo. *Este artigo apresenta técnicas para detecção do protocolo BitTorrent e seus metadados. Foram realizados testes comparando as técnicas de DPI e detecção baseada em assinaturas, utilizando ferramentas open source e comerciais. As capturas de pacotes foram realizadas de diversos sites torrent, com e sem criptografia ativada. As ferramentas escolhidas permitiram realizar a diferenciação entre tráfego legítimo e ilegal.*
Palavras-Chave: BitTorrent, deep packet inspection, assinaturas, IDS, testes.

1. Introdução

Atualmente, o protocolo BitTorrent [Cohen 2008] é amplamente utilizado no mundo pela sua eficiência e facilidade de compartilhamento de arquivos de diversos tipos, tamanhos e para vários fins. Estima-se que 35% do tráfego da Internet é realizado através deste protocolo [Sandvine 2016]. Devido à vasta gama de *downloads* realizados, com conteúdos protegidos por lei sendo distribuídos de forma ilegal [Silva 2015], se faz necessária a pesquisa de técnicas e ferramentas para analisar o conteúdo do tráfego BitTorrent. Considerando que a decodificação deste tipo de fluxo de dados ainda é pouco conhecida e utilizada, devido a sua complexidade, realizar sua identificação é muito importante para prevenir futuros processos judiciais, ocasionados pelo acesso a conteúdos protegidos por direitos autorais. Em virtude do tráfego BitTorrent estar circulando quase totalmente de forma encriptada [Zhe Yang 2012] e devido à ineficácia dos métodos tradicionais de identificação, este trabalho tem por objetivo realizar uma análise das técnicas existentes para detecção e identificação de tráfego BitTorrent.

2. Protocolo BitTorrent

Nesta seção serão apresentadas as principais características do protocolo BitTorrent.

2.1. Funcionamento do protocolo BitTorrent

O protocolo BitTorrent, também conhecido como BTP, é um protocolo *peer-to-peer* (*p2p*) desenvolvido por Bram Cohen, em 2001 [Cohen 2008]. As redes *p2p* se tornaram a principal alternativa na distribuição eficiente de dados na Internet, por terem como característica fundamental o fato de que cada usuário se comporta, simultaneamente, como cliente e servidor. Na Figura 1 pode ser vista a estrutura do arquivo de metadados, gerada a partir da ferramenta dumptorrent [Dumptorrent 2016].

```
root@srv-pasa:~/dumptorrent-1.2# ./dumptorrent -v [otorrents.com]shrek-2001-720p.torrent
[otorrents.com]shrek-2001-720p.torrent:
Name:           Shrek (2001)
Size:           629563832 (600M)
Announce:       udp://open.demonii.com:1337
Info Hash:      13ef3621f73e33edcfda6bc7bcec1221526b1ebf
Piece Length:   1048576 (1M)
Creation Date:  Sat Mar 10 21:48:02 2012
Created By:     uTorrent/3000
Encoding:       UTF-8
Files:
                Other/AhaShare.com.txt                                  59
                Other/Torrent downloaded from Demonoid.com - Copy.txt 47
                Other/Torrent Downloaded From ExtraTorrent.com.txt      353
                Shrek.2001.720p.BluRay.x264.YIFY.mp4                    629319046 (600M)
                Shrek.2001.720p.BluRay.x264.YIFY.srt                    113650 (111K)
                WWW.YIFY-TORRENTS.COM.jpg                               130677 (128K)
Announce List:
                udp://open.demonii.com:1337
```

Figura 1. Estrutura do arquivo de metadados

Para participar de uma rede *p2p* é necessário que seja realizado o acesso a *Sites* públicos ou privados que compartilhem conteúdo através do BitTorrent. Para dar início ao processo de *download* é necessária a obtenção do arquivo de metatados com extensão .torrent, que contém várias informações dos arquivos que serão baixados e dos *trackers* que serão contactados. Na Figura 2 pode ser visto o cliente em contato com o *tracker* e recebendo uma lista com todos os *peers* do *swarm*. Este processo é conhecido como ”anúncio”. Após conectado com os *peers* do *swarm*, o cliente verifica quem tem os pedaços dos arquivos a oferecer. Para cada pedaço recebido, o cliente calcula o *hash* SHA1 do pedaço, conferindo se está de acordo com o valor presente no arquivo de metadados. Após o *download*, o cliente passa a ser um *seeder* dentro do *swarm*.

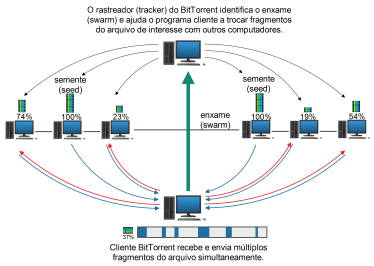

Figura 2. Funcionamento do BitTorrent [HowStuffWorks 2016]

2.2. Criptografia

As técnicas de criptografia *Message Stream Encryption* (MSE) e *Protocol Encryption* (PE) utilizam o algoritmo simétrico de criptografia de fluxo RC4, escolhido pela sua velocidade. O RC4 efetua a troca de chaves combinadas com a *info hash* do arquivo .torrent, visando aumentar a privacidade e confidencialidade. Além disso, elas tentam tornar o tráfego mais difícil de ser identificado por terceiros, incluindo provedores de serviços de Internet (ISPs) e administradores de rede. A encriptação tem por objetivo ofuscar e tornar o tráfego mais difícil de ser detectado, limitando-se a esconder o cabeçalho do protocolo para que não seja possível identificar quem está utilizando o BitTorrent, porém tende a consumir mais recursos do processador, quando ativada.

3. Técnicas de detecção

Nesta seção serão apresentadas duas técnicas de detecção de tráfego, aplicáveis ao padrão de tráfego do protocolo BitTorrent.

3.1. *Deep Packet Inspection* (DPI)

A técnica de *Deep Packet Inspection* (DPI) é baseada na inspeção do *payload* de pacotes, com o propósito de extração de metadados. Pode operar no modo detecção ou prevenção, para proteger redes ou sistemas, de acordo com o posicionamento que for implantado. Em provedores de acesso à Internet, essa técnica é utilizada para proteger as redes internas, bem como as redes dos clientes [Bujlow and Carela-Espanol 2013]. A DPI é utilizada para implementar certas políticas que cobrem as violações de direitos autorais, conteúdo ilegal e utilização abusiva da largura de banda.

3.2. Detecção baseada em assinaturas

Esta técnica visa a detecção por meio de assinaturas e monitoramento das atividades da rede, procurando por eventos que correspondam a padrões pré-definidos de ataques, tráfego malicioso e outras anormalidades. Os eventos correspondentes à determinada assinatura podem ser visualizados em tempo real, através do console *Linux* ou de *interfaces Web* desenvolvidas para esta finalidade. Embora esta técnica seja mais eficiente para identificar tráfego não encriptado do protocolo BitTorrent, ainda existem partes que podem ser identificadas mesmo utilizando a criptografia MSE/PE [Zhe Yang 2012]. As técnicas baseadas em assinaturas são geralmente implementadas em sistemas conhecidos como IDS (*Intrusion Detection System*) e IPS (*Intrusion Prevention System*) [Moraes 2010]. Combinando as melhores funcionalidades de ambas pode-se, além de detectar, prevenir e realizar o bloqueio de atividades maliciosas.

4. Ferramentas

Nesta seção são analisadas ferramentas que implementam técnicas de detecção de pacotes, por assinatura e por DPI.

4.1. nDPI-ntop

É um *software open source* distribuído sob a licença *GNU* LGPLv3. É um sistema multiplataforma e está disponível para instalação em distribuições *GNU/Linux*, sistemas *Windows* e *Mac OS* [nDPI 2016]. A ferramenta suporta identificação de cerca de 170 protocolos. A última versão lançada do nDPI inclui funcionalidades de decodificar e não apenas detectar o tráfego, possibilitando extrair o *hashid* dos arquivos que estão sendo baixados e identificar o conteúdo.

4.2. Xplico

O Xplico é uma ferramenta *open source* de análise forense de rede. Tem por objetivo extrair o conteúdo dos dados, a partir de capturas de tráfego, e possibilidade de reconstruir as conexão de rede, a partir dos pacotes capturados. É um sistema que está disponível somente para a instalação em distribuições *GNU/Linux*, mais especificamente para *Fedora* e *Ubuntu* [Xplico 2016]. A ferramenta possui suporte a uma série de *plugins* que podem "decodificar" o tráfego de rede de diversos protocolos.

4.3. Suricata

O Suricata [Suricata 2016] é um sistema *open source* com funções de IDS (*Intrusion Detection system*) e IPS (*Intrusion Prevention System)*. Implementa uma completa linguagem de assinaturas ligadas a ameaças conhecidas, violações de políticas de segurança e comportamentos maliciosos de *malwares*, *trojans* e outros ataques. Também é uma ferramenta com mecanismos de NSM (*Network Security Monitor*) que realiza o *log* de solicitações HTTP, *logins*, o armazenamento de certificados TLS *(Transport Layer Security)*, a extração de arquivos de um fluxo de dados e o armazenamento em disco. Este tráfego pode ser visualizado em tempo real, na forma de gráficos, através da *interface Web* Snorby [Snorby 2016].

4.4. LANGuardian

O LANGuardian [LANGuardian 2016] é uma solução comercial desenvolvida pela NetFort. É composto por *software* de dectecção de intrusão de rede e possui uma suíte de aplicações de análise de tráfego, tornando-se uma solução única, capaz de detectar anomalias de rede, atividades suspeitas e até mesmo ameaças desconhecidas. Desenvolvida com tecnologia *Deep Packet Inspection*, é capaz de extrair metadados dos pacotes que trafegam na rede. Combinando a técnica de DPI com a de análise baseada em assinaturas, a ferramenta LANGuardian tornou-se uma poderosa suíte de monitoramento.

5. Testes Realizados

Os testes foram aplicados em um ambiente controlado, com o objetivo de avaliar a aplicação das ferramentas e obter dados para análise e comparações.

5.1. Cenário de testes

O cenário de testes foi idealizado sob ambiente físico e virtualizado com VMware Workstation 11 (CPU de 8 núcleos de 3.50GHz e 16GB de memória RAM), contando com cinco máquinas virtuais dedicadas para as ferramentas de análise. Nestas estão instalados os sistemas de monitoramento de nDPI-ntop, Xplico, Suricata, LANGuardian e o *software* hostapd [Hostapd 2016] com funcionalidade de *access point*, por meio de um adaptador USB *wireless* conectado na máquina hospedeira, conforme pode ser visto na Tabela 1.

Tabela 1. Configuração das máquinas virtuais

Softwares	CPU	RAM	HD	Sistema Operacional
nDPI-ntop 1.7	2x3,50 GHz	1GB	60 GB	Debian 7.0
Xplico 1.1.2	1x3,50 GHz	2GB	60 GB	Ubuntu 15.10
Suricata 3.0.1	2x3,50 GHz	4GB	60 GB	Debian 7.0
LANGuardian 14.4.4	2x3,50 GHz	2GB	60 GB	CentOS 7.0
Hostapd 1.0	1x3,50 GHz	1GB	20 GB	Ubuntu 13.10

O hostapd permite a comunicação através de uma *interface wireless* no ambiente de monitoramento de notebooks e smartphones com clientes BitTorrent instalados, conforme pode ser observado na Figura 3.

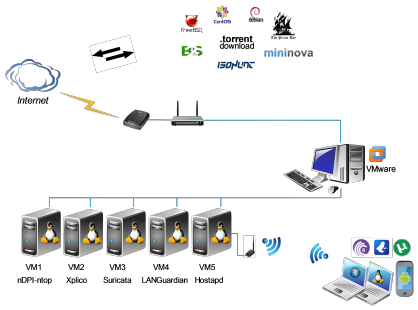

Figura 3. Topologia do cenário de testes

Os clientes torrent selecionados para realização dos testes foram μTorrent, BitTorrent e Vuze. Os dois primeiros foram escolhidos com base em estatísticas de uso pelos clientes torrent de *peers* realizando *downloads*, enquanto o Vuze foi escolhido por ser o

primeiro cliente a implantar a criptografia MSE/PE e por suas diversas funcionalidades integradas [Vuze 2016]. Para realização dos testes, os clientes foram instalados em ambiente misto, físico e virtualizado, em sistemas operacionais *Windows* 7, *Windows* 10, *Ubuntu Desktop* 12 e *Android* 4.1.2.

5.2. Metodologia de Testes

Inicialmente, os testes foram realizados sem criptografia ativada. Estes consistiram na realização de *downloads* de arquivos torrent a partir de *Sites* públicos, tais como isoHunt [isoHunt 2016], The Pirate Bay [Bay 2016], Debian [Debian 2016] e FreeBSD [FreeBSD 2016], bem como de *Sites* privados como o B2S-Share [B2S 2016]. Todos disponibilizam conteúdos de várias categorias através da arquitetura *p2p*.

No estudo realizado por [Sandvine 2015] já havia sido mostrado que no ano de 2015, cerca de 29% dos usuários de BitTorrent utilizavam criptografia ativada em seus clientes torrent (Figura 4a). Para confirmar que esta opção vem se tornando uma crescente entre os usuários de BitTorrent, foram realizados testes para filtrar *peers* que utilizam as opções de criptografia ativada e não ativada. Os resultados podem ser vistos na Figura 4b. Em comparação com 2015, foi possível perceber que houve um aumento considerável no número de usuários que utilizam a criptografia ativada, o que motivou a realização das simulações de captura de metadados utilizando as ferramentas com esta opção.

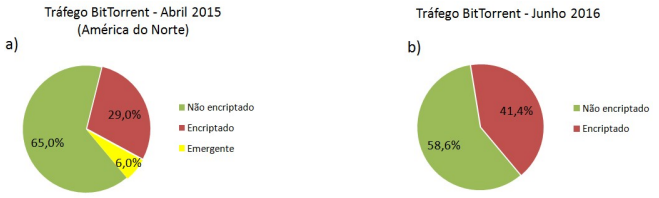

Figura 4. Estatísticas de utilização de criptografia no protocolo BitTorrent

6. Resultados

Após os testes, foi realizada a comparação da eficiência das ferramentas nas capturas de metadados do protocolo BitTorrent, em situações distintas, como pode ser observado no gráfico da Figura 5.

Figura 5. Análise dos metadados detectados

Percebe-se que dois dos metadados, o *announce* e o *info hash*, são identificáveis mesmo que submetidos em testes com criptografia ativada. Estes dados capturados podem ser utilizados como "coringas" na diferenciação de tráfego legítimo de ilegal, a partir de buscas realizadas no Google pelo metadado *info hash*. Outra forma seria relacionar o nome do *tracker* que está presente no *announce* ao nome do arquivo torrent, conforme Figura 6.

Figura 6. Identificação do conteúdo torrent através do *hash* e *announce*

Em praticamente sua totalidade, *hashs* de arquivos torrent disponibilizados em *Sites* públicos são visíveis e identificáveis no buscador do Google [Google 2016], permitindo diferenciar o tipo de conteúdo.

Diferentemente disso, *Sites* de comunidades torrent privadas tais como B2S-Share [B2S 2016] e Manicomio Share [Share 2016], não indexam informações de *info hash* dos seus arquivos torrent em buscadores, dificultando a sua identificação e, por estarem operando em modo privado, somente permitem a comunicação através do seu *tracker* principal, o qual contabiliza a taxa (*ratio*) de *upload* e *download* dos usuários cadastrados. No modo privado não é permitida a descoberta de outros *peers* em um *swarm*. As opções *Local Peer Discover* (LPD), *Distributed Hash Table* (DHT) e *Peer Exchange* (PEX) são desabilitadas por padrão, dificultando ainda mais a detecção dos metadados.

Nota-se que somente as ferramentas LANGuardian e Suricata tiveram a eficácia em detectar os metadados *name* e *files*, possibilitando a identificação do tipo de conteúdo com base nos nomes e arquivos, conforme pode ser visto na Figura 7 e na Figura 8.

Figura 7. Metadados *name* e *file* - LANGuardian

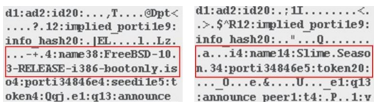

Figura 8. Metadados *name* e *file* - Suricata

As ferramentas nDPI e Xplico não conseguiram detectar os metadados *name* e *files* dos arquivos torrent, mas os campos de *info hash* e *announce* foram detectados, conforme mostra a Figura 9a; porém, estas foram mais eficientes em detectar e apresentar os metadados do campo *announce list*, que se trata dos *trackers* auxiliares, que podem ajudar a identificar a origem do conteúdo torrent. A ferramenta nDPI, em testes com criptografia ativada, teve notável queda de desempenho em detectar o *info hash*, como pode ser visto na Figura 9b.

7. Conclusões

As ferramentas Suricata e LANGuardian foram capazes de identificar e gerar alertas de segurança sobre a atividade do protocolo BitTorrent na rede, bem como a captura de metadados presentes no arquivo torrent. A partir destes dados, foi possível realizar a

Figura 9. Metadados *info hash* **e** *announce* **- nDPI**

diferenciação de tráfego legítimo de tráfego ilegal. A ferramenta Xplico mostrou-se uma ótima solução de análise forense, detectando os metadados e também identificou a atuação de uma empresa que monitora *trackers* de *downloads* ilegais. Entretanto, o Xplico não é destinado a um monitoramento constante de um ambiente de rede. A ferramenta nDPI, além dos metadados, foi capaz de identificar o protocolo QUIC [IETF 2016].

Durante os testes também observou-se que os clientes μTorrent e BitTtorrent implementam um nível de criptografia inferior a do cliente Vuze, que possui opções de utilizar a rede TOR [TOR 2016] e I2P [I2P 2016] para promover o anonimato na Internet. Durante os testes com os clientes *mobile*, notou-se que nenhum deles tem a opção de utilizar criptografia. Pode-se afirmar que a criptografia implementada pelos clientes não é eficiente, a fim de criptografar de forma integral, a comunicação através do protocolo BitTorrent. Conclui-se que atualmente é possível realizar diferenciação de tráfego BitTorrent legítimo de tráfego ilegal, evitando problemas com notificações de direitos autorais.

7.1. Trabalhos futuros

Uma proposta para futuros trabalhos seria um estudo de como realizar o bloqueio do protocolo BitTorrent, com base na classificação e diferenciação de conteúdo legal de ilegal, utilizando ferramentas de IDS/IPS/NSM e verificando a eficiência das ferramentas e bloqueios nesse sentido. Outro trabalho a ser desenvolvido poderia ser o estudo do novo conceito de transferência torrent, o WebTorrent [WebTorrent 2016], que atua de forma híbrida com o protocolo BitTorrent.

Referências

B2S (2016). Available em: <http://www.b2s-share.com/>. Accessed: June 2016.

Bay, T. P. (2016). Available in: <https://thepiratebay.se/>. Accessed: April 2016.

Bujlow, T. and Carela-Espanol, V. (2013). Comparison of deep packet inspection (dpi) tools for traffic classification.

Cohen, B. (2008). Bram Cohen the bittorrent protocol specification. http://www.bittorrent.org/beps/bep_0003.html. Accessed: 2016-03-12.

Debian (2016). Available in: <https://www.debian.org/>. Accessed: June 2016.

Dumptorrent (2016). Available in: <https://sourceforge.net/projects/dumptorrent/>. Accessed: April 2016.

FreeBSD (2016). Available in: <https://www.freebsd.org/>. Accessed: June 2016.

Google (2016). Available em: <https://www.google.com.br/>. Accessed: June 2016.

Hostapd (2016). Available in: <https://github.com/jenssegers/RTL8188-hostapd>. Accessed: March 2016.

HowStuffWorks (2016). Available in: <http://tecnologia.hsw.uol.com.br/bittorrent.htm/>. Accessed: April 2016.

I2P (2016). Available em: <https://geti2p.net/ >. Accessed: June 2016.

IETF (2016). Available em: <https://tools.ietf.org/html/draft-tsvwg-quic-protocol-00/>. Accessed: June 2016.

isoHunt (2016). Available in: <https://isohunt.to/>. Accessed: April 2016.

LANGuardian (2016). Available em: <https://www.netfort.com/languardian/>. Accessed: April 2016.

Moraes, A. F. (2010). *Segurança em Redes - Fundamentos 1. ed.* Editora Érica Ltda, São Paulo.

nDPI (2016). Available em: <https://github.com/ntop/nDPI/>. Accessed: April 2016.

Sandvine (2015). Available in: <https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/encrypted-internet-traffic.pdf>. Accessed: June 2016.

Sandvine (2016). Available in: <https://www.sandvine.com/downloads/general/global-internet-phenomena/2013/sandvine-global-internet-phenomena-report-1h-2013.pdf>. Accessed: August 2016.

Share, M. (2016). Available in: <https://www.manicomio-share.com/>. Accessed: June 2016.

Silva, R. (2015). Relatório indica que brasil é campeão do mundo em pirataria de séries. https://tecnoblog.net/172165/brasil-campeao-pirataria-series/. Accessed: 2016-03-17.

Snorby (2016). Available em: <https://github.com/Snorby/snorby/>. Accessed: April 2016.

Suricata (2016). Available in: <https://suricata-ids.org/>. Accessed: March 2016.

TOR (2016). Available em: <https://www.torproject.org/>. Accessed: June 2016.

Vuze (2016). Available in: <http://www.vuze.com/>. Accessed: April 2016.

WebTorrent (2016). Available em: <https://webtorrent.io/>. Accessed: June 2016.

Xplico (2016). Available in: <http://www.xplico.org />. Accessed: April 2016.

Zhe Yang, Lingzhi Li, Q. J. (2012). Zhe Yang, Lingzhi Li, Qijin Ji cocktail method for bittorrent traffic identification in real time. http://www.ojs.academypublisher.com/index.php/jcp/article/view/jcp07018595. Accessed: 2016-04-15.