5T-01

シグネチャ法を用いたHTTP通信のパケット高速検知の提案

功刀 剛† 小倉 加奈代† Bhed Bahadur Bista† 高田 豊雄†
岩手県立大学ソフトウェア情報学部ソフトウェア情報学科†

1. はじめに

ブロードバンド環境の普及により，大容量高品質データの送受信が容易となった．一方で，映画やゲームソフトなどのデジタルコンテンツが違法にダウンロード(以下 DL)されるようになり大きな問題となっている.フランスでは対策として P2P による違法 DL を検知するシステムを運用し，違法者に警告や罰金を科している[1].しかし Giblin の調査[2]によると，実際に減少しているのは監視が可能な P2P での違法 DL のみで， HTTP 通信での違法 DL は増えていることが報告されている．

その問題を受けアメリカでは，全てのパケット通信を監視して違法 DL と思われる通信を検知するシステムを運用した.しかし合法サイトからの DL も違法 DL として検出してしまい，本来の目的を果たすことができず現在ではシステムを休止している.本研究ではそれらの問題を解決するために，インターネット上にある違法ファイルを監視して違法 DL 検知を行うシステムを提案する．

2. 背景の考察

2.1 シグネチャ検索法

本研究では違法 DL 検知に有川らが提案したシグネチャ法[3]を用いて行う.シグネチャ法とは 主に文書検索において使われる検索方法で，高速 に検索文字が入っている文書を探し出す方法である．

①検索前の処理として文書内にある単語を全て 抜き出しハッシュ化を行い，単語シグネチャを生成する．

②生成された全ての単語シグネチャを論理和で計算し文書 シグネチャを生成する.(文書ごとに入っている 単語が違うため，文書ごとに固有の文書シグネチャが生成される)

③検索時の処理として検索文字をハッシュ化して検索シグネチャを生成する.(単語シグネチャと同じハッシュ関数を利用するため，同じ単語であれば同じシグネチャが生成される)

④生成した検索シグネチャと文書シグネチャを 比較する.文書シグネチャの中に検索シグネチャが内包されていれば，検索単語がその文書内に 入っている可能性を判断できる．

ここでは i,j∈{0,1}が i≥j であるとき，i は j を内包するといい，記号 i>j と表す．

同じ長さの 2 進ビット列 i,j ですべてのビット位置が内包する時 i は j を内包するという．

シグネチャ法のメリットは検出漏れがないことと高速に検索が可能である点が挙げられる.しかしシグネチャ法で判断できることは，文書内に検索文字がある可能性だけである.そのため誤検出が起きてしまうため，シグネチャ法での検索の後に誤検出が起きているか確認する必要がある．

2.2 パトリシアトライ木によるシグネチャ検索

前述のシグネチャ法の問題点として探索コストの増大が挙げられる．シグネチャ法では検索のために使われる文書シグネチャが検索対象文書ファイルごとに作成される．そのため検索対象の文書ファイルを全て検索するためには，作成された文書シグエネチャの数だけ比較を行う必要がある．この様な比較方法では計算時間が検索対象ファイルのファイル数に比例してしまうため効率が悪くなってしまう．

本研究では探索コストの削減のために権藤らが提案した高速テキスト検索のためのパトリシアトライ構造化シグネチャファイルを用いる．この探索方法はパトリシアトライ木を利用して，効率よくシグネチャ検索を行う手法で以下の手順で探索を行う (図 1)

①検索前の処理として文書シグネチャを元にパトリシアトライ木を構築し探索木を作成する．(本研究での探索木は，葉に文書シグネチャデータが格納され，節点には左右の枝の共通部分データを格納する)

②探索の処理として最上位の節点から探索を行う．節点のスキップ数を確認し「スキップする部分シグネチャ」＜「それに対応する問い合わせシグネチャ」が真となる時③のステップへ進み，偽である場合には探索を破棄する(節点のスキップ数が 0 の時は真と判断して③のステップへ進む)．

③問い合わせシグネチャの比較済みビットを確認し，その次のビットが「1」であるならば両方の枝，「0」であるならば左の枝を探索する．そして枝の先の節点データを元に②の処理を行う．

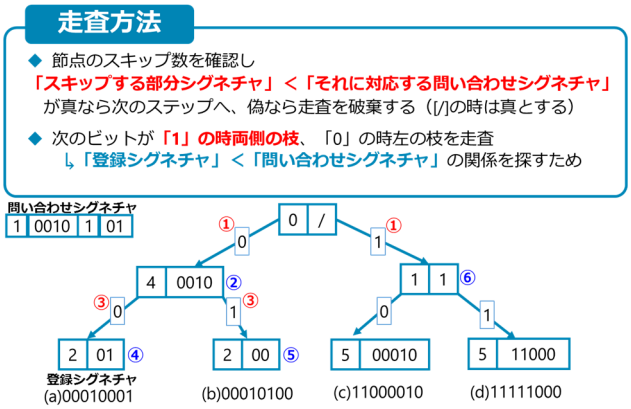


図 1. パトリシアトライ木での走査方法

Proposal of high speed packet detection for HTTP communication using signature method
Tsuyoshi Kunugi, Ogura Kanayo, Bhed Bahadur Bista and Takata Toyoo
Faculty of Software and Information Science, Iwate Prefectural

3. 検知システム

3.1 シグネチャ検出方法

本提案でのパケット検出方法の概要を図2に示す.パケット検出のためには通信されたパケットを採取してパケットストリームを組み立てる必要がある.しかしそのようなキャプチャリングでは処理手順が多く，保存領域にも大きなリソースが要求される.そこで本提案では前述のシグネチャ法[3]を使い，パケットストリームを組み立てることなくパケットを検出する.(図2)

①検索前の処理として検知したいファイルデータの末尾 64Byte をシステムに登録する(登録シグネチャ).

②ISP が管理しているルータからパケットを採取する.この時,TCP の ACK パケットや,DNS への問い合わせパケット等も区別せずに採取する.

③システムは「(a)キャプチャしたパケット」と「(b)パケットの末尾 64Byte のデータ」の 2 つを各バッファに保存する.(b)のデータ全ての論理和を計算し，パケットシグネチャを作成する.

④バッファ内のデータが一定数溜まったところで,登録シグネチャとパケットシグネチャを比較する.パケットがあると判断された場合はデータを保存し,無い場合は破棄する.保存したパケットを詳しく照合し,ダウンロードパケットを検出する.

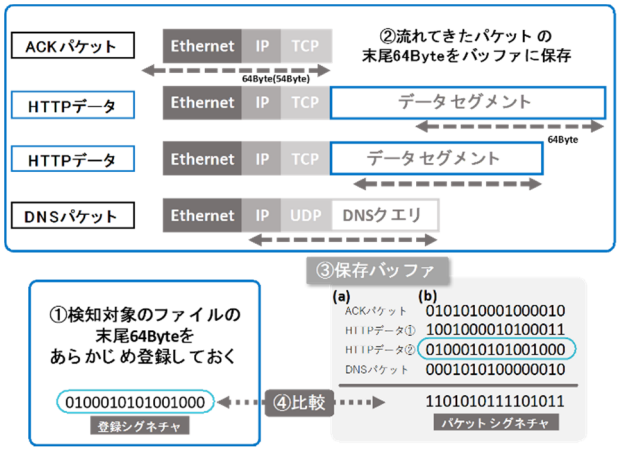


図 2. パケット検知の流れ

4. システム評価

4.1 実験内容

提案手法が想定システムにおいて，既存手法より高速かつ誤検知がない手法であることを示すため以下の条件で実験を行う.

①オンラインストレージ(アップローダ)にテスト用ファイルをアップロードする.このファイルをDLした時のパケット通信を PC 上でキャプチャしておく(35527 パケットの通信データ).

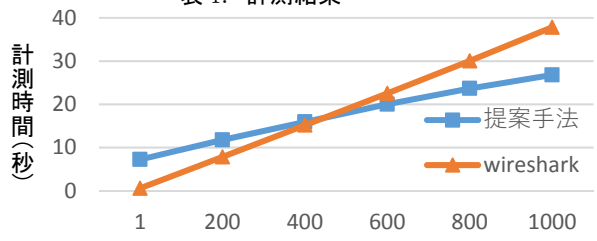
②キャプチャした通信データとテスト用ファイルを元にパケット検索を行い，既存手法と提案手法の計測時間を比較する.

③想定システムでは，検知対象の違法ファイルが運用時間と共に増えていくことが考えられる.そのため検知したいファイルデータとして，テスト用ファイルとダミーファイルを用意する.ダミーファイルの数を徐々に増やし，それぞれの計測時間を測る.ダミーファイルの内容は ランダムなバイナリデータとする.

4.2 実験結果

既存ツールとして Wireshark と Tcpdump を用いた.Tcpdump では grep コマンドを利用して検索を行った.

表 1. 計測結果



	1	200	400	600	800	1000
提案手法	7.2322	11.762	15.925	20.011	23.654	26.781
Wireshark	0.5628	7.8266	15.177	22.539	30.052	37.829
Tcpdump	0.069	57.492	113.39	170.49	228.35	289.25

計測を行った結果，同時検索ファイル数が 400 より多い時，提案手法の方が高速に検知を行える ことが分かった.このような結果になった理由 として以下の点が推察できる.

・シグネチャ法を用いてパケットをまとめて検索したため，計測時間の伸びが既存ツールより緩 やかになった.

・検索前の処理としてファイルデータから登録 シグネチャを生成するため，検索ファイル数が 1 ファイルの時約 7 秒要した.

5. 終わりに

本研究では，シグネチャ法を用いた HTTP 通信による違法DL検知システムを提案した.提案手法の有効性を示すことができたが，提案したシ ステムには考えらえる攻撃方法がいくつか存在 している.今後はそれらの攻撃方法の対策を考慮 しシステムの改善を行う.

参考文献

[1] Hadopi | Haute Autorité pour la diffusion des œuvres et la protection des droits sur internet (online), available from <https://www.hadopi.fr/>(accessed 2017-06-15).

[2] Giblin, R.: Evaluating Graduated Response, Columbia Journal of Law & the Arts, vol.37, pp.147-210(online), available from <http://dx.doi.org/10.7916/D8F47M3W> (2013).

[3] 有川節夫, 篠原武, 松本一教, 張裕民, “重ね合わせ記号を用いた文献検索システムについて －キーワードのための重ね合わせ符号－”, 情報処理学会研究報告データベースシステム(DBS), 48(1986-DBS-054), pp.1-8, 1986.

[4] 権藤夏男 ， 金子邦彦 ， 牧之内顕文, “高速テキスト検索のためのパトリシアトライ構造化シグネチャファイル”， 情報処理学会研究報告, 97 巻 64(DBS-113) ,pp.191-196, 1997.