Vol.2009-CSEC-46 No.15
2009/7/2

不適切な HTTP トンネリング通信を検出する手法の提案

鳥居 悟[†]　　小村 昌弘[††]

不注意によるものを含む，組織内部を発端とするセキュリティ侵害への対応が今後重要になってくると考えられている．本論文では，規程に反するサービスの使用状況の実態を把握することを目的に，ネットワークの挙動を監視することでとで，HTTP プロトコルを使用した VPN やファイル共有を行う通信，すなわち，不適切な HTTP トンネリング通信を検出する手法について報告する．
提案する手法は，HTTP プロトコルに特化した不正通信の検知技術である．本手法の特徴は，すでに不正プログラムが組織内に存在しているとの前提をおき，組織内の複数クライアントの通信挙動を解析することで，トンネリング通信を行うアプリケーションの特定とその特徴を抽出できることである．

Proposing the method for detecting an improper HTTP tunneling communication

TORII, Satoru[†]　KOMURA, Masahiro[††]

It is thought that doing measures against the security breach of which the beginning is the inside of the organization will become important in the future. The purpose of this study is to understand the realities of the usage condition of service in contradiction to the rule. In this paper, it reports on the technique for detecting the HTTP tunneling communication that does VPN and the file sharing using HTTP protocol by observing behavior on the network.
The method for proposing it is a detection technology of an illegal communication that specializes in HTTP protocol. The feature of this method are as follows; assumption that the malicious program has already existed in the organization is put, the communication behavior of two or more in organization clients is analyzed, and specific and the feature of the application to communicate tunneling should be able to be extracted.

1. はじめに

組織内ネットワークにおいてはすでに，ファイアウォールやアンチウィルスソフトなど，さまざまなセキュリティ対策が施されている．一方で，このようなセキュリティ対策を回避するソフトウェアも数多く出回っている．例えば，マルウェア特にボットは，環境に応じて動作を変更するなど巧妙になってきている．また，ファイアウォールで許可されているネットワークプロトコルを用いて，制限されているサービスを使用可能とするトンネリング通信ソフトが気軽に使用できるようになっている．

これらのソフトウェアにより，今まで構築してきたセキュリティ対策に対する投資が無駄になる可能性がある．利用可能な通信プロトコル (HTTP 等) を用いて，リスクのある通信プロトコルが使用できてしまうなど，フィルタリング機能が回避されてしまうことになる．近年は，このようなファイアウォールを迂回する通信を行うアプリケーションが増加しており，特に，クリックひとつでインストールできるなど，利用者がその脅威を意識することなく簡単に使うことができてしまう．たとえ便利な道具であっても，組織内の利用者が安易に利用することは，組織においては重大な脅威となる．

本研究は，ネットワークを監視し，観測された通信データを解析することで，トンネリング通信を行うアプリケーションの特定とその特徴(パケット数やバイト列など)を抽出することを目指すものである．

2. HTTP トンネリング通信

2.1 組織における脅威

トンネリング通信[1)]とは，他のプロトコルのパケットにカプセル化して，本来行いたい通信を行うものである．特に，HTTP トンネリング通信においては，組織において許可されている HTTP を用いてカプセル化することにより，許可されていない通信が行えるものとなる．

そのため，このような通信は，組織にとっては以下のような脅威が生じる．

- ファイアウォールのフィルタリング機能が回避されてしまう

利用可能な通信プロトコル (HTTP 等) を用いて、リスクのある通信プロトコルが使用できる

- 出入り口でのウィルス対策が回避されてしまう

通信が暗号化されており、ウィルスチェックできない

† 株式会社富士通研究所
Fujitsu Laboratories Ltd
†† 富士通株式会社
Fujitsu Limited

- 機密情報が漏洩してしまう
  当事者も気付かぬ間に、PC 上の業務データが発信される
- ウィルスやワームに感染/蔓延してしまう
  外部から能動的に、(感染した) ファイルを送りつけることができる
- 利用者がその脅威を認識せずに使用している
  インストール不要なものもあり、誰でも簡単に利用できる

2.2 HTTP トンネリング通信方式

HTTP のパケットに他のプロトコルをカプセル化するには，大きく以下の三つの方法がある．

(1) **GET/POST 型**

HTTP の GET メソッドと POST メソッドを用いて，データのやり取りを実現することができる．すなわち，POST メソッドを用いてクライアント側からデータを送出し，GET メソッドを用いてデータを受け取ることができる．

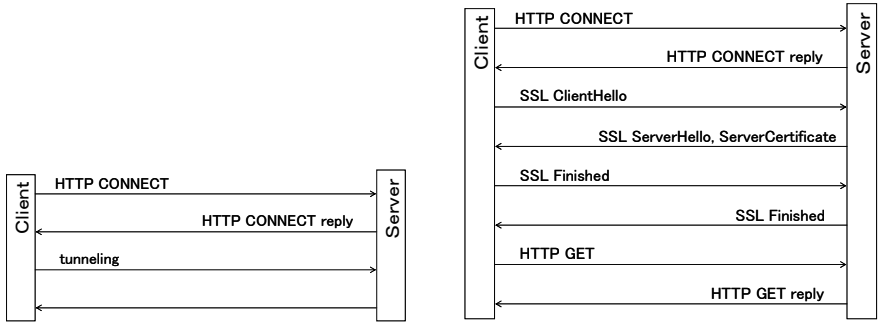
(2) **CONNECT 型**

HTTP 1.1 から CONNECT ヘッダが規定され，持続的接続が可能となった2)]．これにより，サーバからデータを取得せずにセッションの確立だけを行い，その後の通信では HTTP プロトコルに限らずあらゆるプロトコルを流すことが可能となる(図 1(1))．

(3) **SSL 型**

HTTP ポートに SSL プロトコルを流す場合，SSL プロトコルを HTTP プロトコルにカプセル化するのではなく，CONNECT リクエストを利用する．すなわち，CONNECT リクエストによりセッション確立が行われた後，SSL プロトコルに基づくやりとりを行う[3)]（図 1(2)）．

クライアントとサーバの間でパケット転送をするプロクシーサーバーやファイアウォールは，途中 SSL プロトコルのパケットが流れるが，HTTP プロトコルの規約に従ったパケットが流されていると解釈する．



(1) CONNECT 型　　(2) SSL 型

図 1：HTTP トンネリング通信方式

近年の HTTP プロトコルを使用した VPN やファイル共有を行うアプリケーションは，主に(2)CONNECT 型と (3)SSL 型のどちらかで実現していると考える．そこで，本研究ではこの二つの手法を検知対象とする．

2.3 組織としての対応

組織において，HTTP を止めることはその弊害の方が大きく現実的ではない．また，各 PC の管理者権限の剥奪や，アプリケーションのインストールを一切禁止にすることは，セキュリティ対策のひとつとして有効ではあるが，IT 利活用のイノベーションを阻害する要因となりうるので，慎重な対応が望まれると考える．そのため，多くの組織では，運用管理部門が様々なアプリケーションを調査し，その脅威が明確になったものを特定し，事前に内部規定で「禁止」と定めている．

しかしながら，運用管理部門が気付かない危険なアプリケーション/サービスが他にもあるかもしれず，妥当性や網羅性に課題があった．使用禁止が遵守されているかなど，現状のセキュリティ対策の遵守状況を把握しておくことが必要である．

すなわち，このような内部規程に反するサービスの使用状況の実態を把握する，組織内システムの脅威を未然に防ぐことを目的とした，検知手法の確立が求められている．

3. 関連研究

不正な通信を検知する技術は，これまで外部の悪意を持った者からの通信を検知することに主眼がおかれていた．従来の検知技術としては，シグネチャ型検知とアノマリ型検知の大きく二つに分類できる．

(1) **シグネチャ型検知[4)]**

「侵入行為は形式的に表現できる」という概念を基本とするものである．既知の攻撃パターンによるものは検知可能であるが，未知の攻撃パターンのものの検知はほとんど不可能である．この攻撃パターンを作成するためには，不正プログラムなどを入手し，その攻撃手法を解析することが必要である．予め不正プログラムなどを特定・入手しなければならず，未知の不正プログラムに対して検出するのは困難である．

HTTP トンネリング通信を行うアプリケーションが一度特定できれば本技術が利用可能であるが，管理者も知りえない様々なアプリケーションの不正使用を検知するには不充分である．

(2) **アノマリ型検知[5)]**

「侵入行為は通常行為とは異なる」という概念を基本とするものである．システムにおける通常行為に反するものは侵入行為とみなす．不正と正常とを切り分ける閾値の算出が困難であり，また，検知された事象が具体的にどのような脅威であるのかを見極めるのが困難であった．

不正と正常とを切り分ける様々な取り組みも行われている．例えば，あらかじめ解析の前に「正常なデータ群」と「異常なデータ群」を用意し，両者の特徴を分析し双方の差異から境界を規定する．検知対象データがどちらに近いかで判断することが行われている．

すでに不正プログラムが組織内に存在しているとの前提のもとで，このような挙動を検出するには，正常なデータ群をどのように定義するかが課題となる．必ずしも，当該組織の過去の状況や他組織の状況を正常な状態として定義してよいとは限らない．また，異常なデータ群の入手にあたっては，結局のところ，予め不正プログラムなどを特定することが求められ，シグネチャ型検知と同様の課題が発生する．

正常なデータ群として「標準的なふるまい」を設定し，これに適合しないものを異常と判定する手法がある．例えば，グループ分けを行いこれらのグループに属さない少数派を抽出する，観測されるデータの想定範囲を規程しその想定範囲に外れるものを抽出する，などである．

本提案手法は，このやり方に近い．すなわち，組織内の大部分は内部規程に準じて正しい振る舞いをしているとの仮定をおき，観測されたデータの大部分は正しい挙動であり,大多数とは異なる挙動は内部規定に違反している可能性があるのではないか，と考える．

4. 検知手法

本研究は，ネットワークを監視し，観測された通信データを解析することで，トンネリング通信を行うアプリケーションの特定とその特徴(パケット数やバイト列など)を抽出することを目指すものである．

本提案手法は，HTTP プロトコルに特化した不正通信検知であり，組織内の複数クライアントの集団的行動の傾向をみることで，不正なアプリケーションの通信を抽出するものである．すなわち，すでに不正プログラムが組織内に存在しているとの前提のもとで，従来の不正プログラムを入手・解析することで抽出した検知に必要な特定のパターンを，流れている通信を入手・解析し正常な通信と切り分けることで，効率よく抽出するものである．

そのためには，以下の三点を明らかにすることが必要と考える．

(1) **正常な通信とトンネリング通信とうまく切り分けられるか**
(2) **抽出されたトンネリング通信から、その検知のための特徴が抽出できるか**
(3) **当該通信を発信するアプリケーションを特定できるか**

4.1 正常な通信とトンネリング通信との切り分け

4.1.1 アクセス先 URL

組織内の様々な利用者は様々な Web サイトにアクセスする．すなわち，Web サイトからみると，様々な利用者からアクセスされていることになる．一方で，トンネリング通信の使用者は限定的であり，そのアクセス先も限定的であると考えられる．あるアクセス先 URL にアクセスするクライアントを調査することで,様々なクライアントからアクセスされている場合は普通の Web サーバとみなすことができ,一部のクライアントからしかアクセスされていない場合はトンネリング通信のアクセス先とみなすことができると思われる．

このように，セッションをアクセス先 URL ごとに分類することで，セッションの様子に違いが現れ，Web ブラウザとそれ以外のソフトウェアの区別がつくと思われる．

4.1.2 User-Agent

一般的な Web ブラウザの場合，User-Agent に格納される文字列は，Mozilla で始まりそのうしろには OS のバージョン，パッチ適用状態，モジュールの名前，版数などが記載される．それらの情報を組み合わせた User-Agent の種類は数百種にもなる．

例えば，アクセス先の種類を見ることで，特定のアクセス先にだけアクセスしているものを見つけることが可能となる. Web ブラウザであれば数多くの Web サーバにアクセスするので，このような User-Agent を名乗るクライアントソフトウェアは，Web ブラウザでない可能性がある．

セッションを User-Agent ごとに分類することで，セッションの様子に違いが現れ，Web ブラウザとそれ以外のソフトウェアの区別がつくと思われる．

4.1.3 SSL 暗号方式

SSL セッションが開設されるときに送信される Client Hello には, クライアントソフトウェアがサポートする暗号方式のリスト(Client Cipher Suite)が含まれている. この暗号方式のリストは，クライアントソフトウェアの実装によって異なるものである．

セッションを SSL 暗号方式ごとに分類することで, Web ブラウザとそれ以外のソフトウェアの区別がつくと思われる．

4.2 検知のための特徴抽出

組織内の様々な利用者は様々な Web サイトにアクセスするため，一般的な HTTP 通

信では，様々なデータサイズの，様々なデータ内容が流れるものと思われる．

一方，通信セッションを継続することが多いトンネリング通信では，セッションを開設するためのネゴシエーションや，セッションを切断されることなく維持するためのハートビート，処理要求を示す制御コードなどが流れる．そのため，トンネリング通信のセッションには，以下のような特徴があると考えられる．

- 特定のバイト数のパケットがある
- 特定のバイト列がパケット中に現れる

また，HTTP トンネリング通信に限らず，Web アクセスのフィルタリングなどで採用されているアクセス先 URL 特定や，侵入検知ツールなどで採用されている User-Agent に含まれる特定文字列なども，検知のための特徴として有効と考えられる．

4.3 アプリケーションの特定

抽出された怪しいセッションの通信ログに含まれている情報を基に，アプリケーションの特定を試みる．通信ログに含まれている情報としては，以下のものがある．

- アクセス先：アクセス先の URL．CONNECT リクエストのあて先．
- User-Agent：ソフトウェアが User-Agent をつけるかは任意であり，詐称も可能であるため，わからない場合や正しくない場合もある．
- 送信元：クライアントの IP アドレス．プロキシが追加する．追加するのは必須ではないのでない場合もある．
- その他パケット中に含まれる可読文字列

第一ステップとしては，アクセス先 URL に記載されている，IP アドレスや FQDN から，そのサイトの所有者を whois[6]などで調査する．

第二ステップとしては，これらの入手できた可読文字列を基に，検索エンジンを用いて調査する．アプリケーション所有者が開設している公開 Web サイトや，所有者が提供しているサービスやソフトウェアの情報を掲載しているサイトが抽出できたならば，所有者やソフトウェアの情報を入手する．

第三ステップとして，これらの作業で判明しない場合，通信を検知した日時にどのようなソフトウェアを使用していたか，当該通信の送信元に問い合わせる．

現時点では，これらの手順はツール化されておらず，人手による作業に依存する．

5. 実証実験

5.1 試行環境

本検知手法の評価を行うための実証実験を行った．組織内とインターネットとの接続口における通信データを収集し，その通信データに対して本提案手法に基づく解析を行い，トンネリング通信の特徴が抽出できるかどうかを評価した．

観測対象のネットワーク構成は，組織内利用者が外部 Web サーバにアクセスするためのプロキシがファイアウォールの内側に配置されているという，組織における一般的な構成となっている（図 2）．

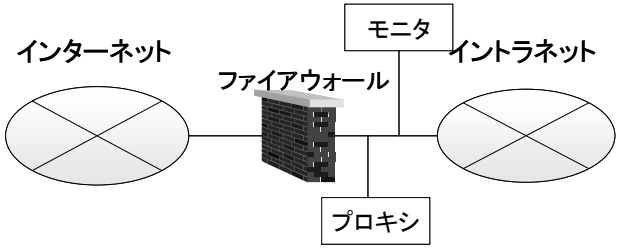


図 2：試行環境

5.2 取得した通信データ

本実験で使用したのは，2008 年 7 月 2 日の朝 6 時から夜 12 時までの 1 日分の通信データである．取得した通信データから通信フローを再構築し，接続が継続する一連の HTTP 通信をひとつのセッションとした．取得した全通信データを再構築した通信セッションの内訳は，図 3 に示すとおりである．

今回の解析対象は，CONNECT ヘッダを含むセッションとし，サーバ・クライアント間でそれぞれ 10 回以上のやり取りを行ったものとした．トンネリング通信は，接続を継続し異なるプロトコルのやり取りを行うものであり，この解析対象は妥当なものと考える．

5.3 試作した検知システム

本実験で試作した検知システムの構成を図 4 に示す．tcpdump[7)]で取得した通信パケットを入力とし，通信セッションを再構築し，分類し不振なセッションを抽出する．アプリケーションが特定でき，その検知パターンが確立できた場合には，検知用のパターンに登録する．

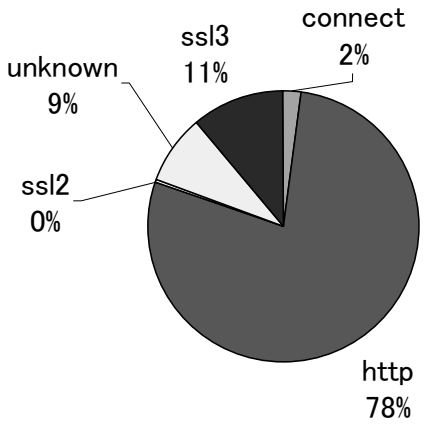


図 3：通信セッションの内訳

一方，同様に，tcpdump で取得した通信パケットを入力とする検知モジュールがあり，登録された検知用パターンに基づき，該当するアプリケーションの通信を検知し，その結果を通知する．

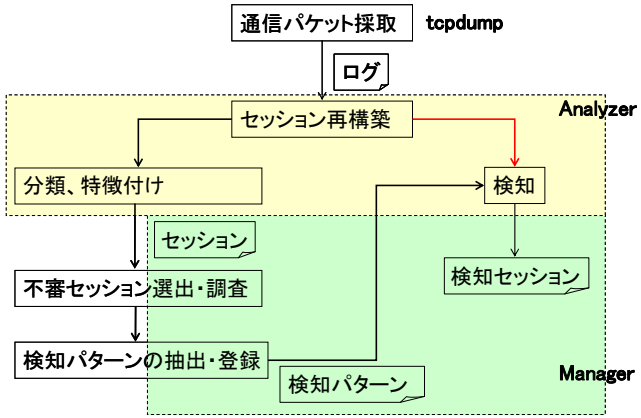


図 4：試作した検知システム

5.4 実験結果

5.4.1 分析結果の全体像

解析対象のセッションにおける CONNECT 型，SSL 型のそれぞれセッション数は、表 1 の通りである．なお，この数値は今回の解析対象のものであり，CONNECT ヘッダを含むセッションとし，サーバ・クライアント間でそれぞれ 10 回以上のやり取りを行ったものに限定している．

表 1：解析対象のセッション数

	総セッション数
connect型通信	2,226
SSL型通信	198,134

本来，connect ヘッダは，通信を HTTP プロトコルから SSL プロトコルに変更するときに用いられるものと考えられる．すなわち，先の表＊における connect 通信型に分類される 2,226 セッションはすべて，疑義ある通信であるとみなすことができる．

5.4.2 正常な通信とトンネリング通信との切り分け

さらに，通信セッションに対してそれぞれの特徴を抽出し，不適切なセッションの抽出を行った．ここでは，SSL 型通信に該当する通信セッションを，アクセス先 URL，User-Agent，SSL 暗号化方式ごとに集約し，それぞれの集約された通信セッション群の特徴を整理した．

(1) **アクセス先 URL**

図 5 は，アクセス先 URL ごとに集約した，セッション数の多い上位 50 件のセッション群に関して整理したものである．上から，セッション数，User-Agent の種類数である．横軸はアクセス先 URL，縦軸はそれぞれの種類数値である．

図中，セッション数のわりに，種類数が少ないアクセス先 URL が多く見られる．一般的な Web サーバへは様々なクライアントからアクセスされるとの仮説に基づけば，これらの User-Agent の種類数が少ないアクセス先 URL はトンネリング通信のアクセス先との疑いがある．

同様に，図 6 は，User-Agent ごとに集約した，セッション数の多い上位 50 件のセッション群に関して整理したものである．上から，セッション数，アクセス先 URL の種類数である．

図中，アクセス先 URL の種類数が少ない User-Agent が散見される．特定のアクセス先にだけアクセスしている User-Agent であり，トンネリング通信の疑義がある．

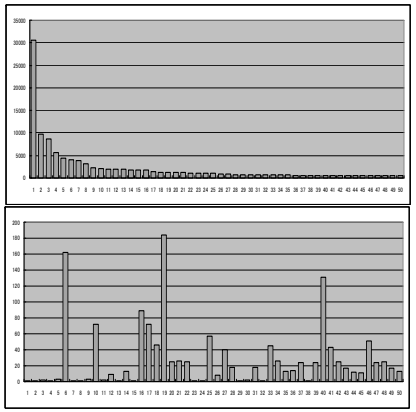


図 5：アクセス先 URL の集約結果

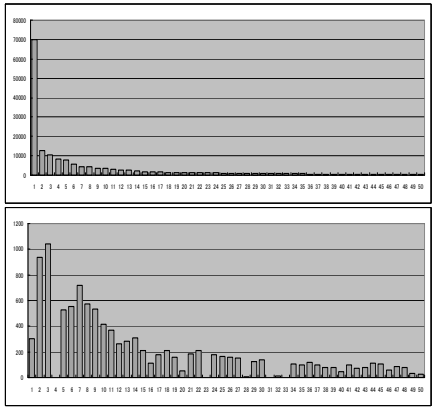


図 6：User-Agent の集約結果

図 7 は，SSL 暗号方式ごとに集約した，セッション数の多い上位 50 件のセッション群に関してそのセッション数を表したものである．

他のグラフとは異なり，特定の SSL 暗号方式にセッション数が集中している．当該 SSL 暗号方式を実装しているアプリケーションは，多くの通信を行っている一般的なものと捉えられる．一方これ以外の SSL 暗号方式を実装しているアプリケーションは少数派であり，大多数のものが正しい挙動との仮定に基づけば，これらはトンネリング通信の疑義がある．

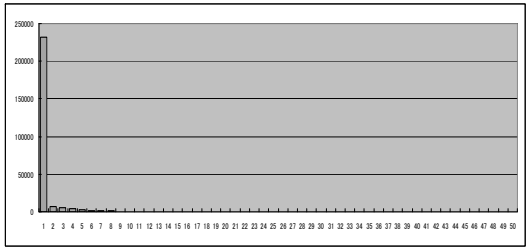


図 7：SSL 暗号方式ごとのセッション数

5.4.3 検知のための特徴抽出

ひとつの通信セッションにおいて，同じバイナリ列が何度も現れるもの，同じサイズのパケットが何度も現れるものは，怪しい通信の可能性がある．すなわち，トンネリング通信に特有の通信セッションを継続するなどのための制御情報をやり取りしている可能性が高い．そこで，以下の手順で特定のバイト数のパケットやバイト列の除法を取得した．

(1) **特定のバイト数のパケット**

通信セッションに含まれるパケットに対して，それぞれのパケット長をセッション毎に集計しておく．同じ長さのパケットが規定数以上含まれていた場合に，そのパケットサイズ・回数と共に当該通信セッションに関連付けて記録する．異なるパケットサイズのものが複数出現する通信セッションもありうる．

(2) **特定のバイト列が現れる複数パケット**

通信セッションに含まれるパケットに対して，それぞれのパケットに出現するバイナリ列を記録しておき，当該通信セッションの他のパケットにおいて，そのバイナリ列が出現するかどうか照合する．主に，制御情報はパケット中の特定の位置に配置されるケースが多いので，バイナリ列が出現する位置は重要である．

5.5 アプリケーションの特定

本実証実験では，アプリケーションの特定は人手で実施した．解析対象の全てのセッションに対しては実施できず，疑義ある通信セッションに対してのみ実施した．その結果, 4,047 セッション, 64 種類のソフトウェア/サービスを特定することができた．

5.6 考察

実環境における通信データを基に行った実証実験の結果, connect 通信型に分類される通信セッション，一部のクライアントからしかアクセスされていないアクセス先 URL，特定のアクセス先にだけアクセスしている User-Agent，一般的ではない SSL 暗号方式を実装しているアプリケーションなど，トンネリング通信とみなせる疑義ある通信を抽出することが出来た．また，人手で実施したアプリケーションの特定作業により，具体的なソフトウェア/サービスを特定することができた．

これらのことから，本提案手法が実環境において有効であるとの感触が得られた．

6. おわりに

本論文では，ネットワークを監視し，観測された通信データを解析することで，トンネリング通信を行うアプリケーションの特定とその特徴（パケット数やバイト列など）を抽出することを目的に，その分析手法の検討を行った．組織内ネットワークに

Vol.2009-CSEC-46 No.15
2009/7/2

おける実証実験を行った結果，トンネリング通信とみなせる疑義ある通信を抽出することが出来でき，具体的なソフトウェア/サービスが特定できた．

本研究におけるトンネリング通信は，CONNECT リクエストを用いた HTTP 通信を対象とした．既存の Web ブラウザをそのまま使用するものなど，トンネリング通信の実現手段が他にもあると考えられる．また，本技術の適用条先は，大部分の利用者はトンネリング通信を行っていない組織である．

本論文で提案した手法は，トンネリング通信の検知以外にも利用可能と考える．すなわち，組織内ネットワークにおける不正通信検知，SPAM メールを検出するレピュテーション，ISP 環境下におけるボット感染 PC の検出，などに適用可能と考える．

今後は，本実証実験結果を理論的に検証すると共に，他の不正通信の検知へ適用範囲の拡大化を検討していきたい．

参考文献

1) JakeHill, “Bypassing Firewalls: Tools and Techniques,” March 2000
2) Fielding, et al. ”RFC2616 - Hypertext Transfer Protocol -- HTTP/1.1,” June 1999
3) Rescorla, “RFC2818- HTTP Over TLS,” May 2000
4) Stefan Axelsson, “Intrusion Detection Systems:A Survey and Taxonomy,” March 2000
5) Chandola, et al. “Anomaly Detection: A Survey,” To appear in - ACM Computing Surveys, A modified version available as a technical report at: http://www.cs.umn.edu/tech_reports_upload/tr2007/07-017.pdf
6) Daigle, “RFC 3912 - WHOIS Protocol Specification,” September 2004
7) http://www.tcpdump.org/