動的ファイアウォールシステムのための DNSによるクライアントIPアドレス通知機能

大塚 友和[1]　ガーダ[1]　山井 成良[2]　岡山 聖彦[3]

概要：近年，組織外から組織内の計算機に対する不正アクセスが後を絶たず，その対策は急務である．対策の一つとして，ファイアウォール製品が用いられている．ところが，既存のファイアウォール製品は管理者が手動で設定を行う必要があり，またあらかじめ通信相手として認識しているものしか設定できないという問題がある．本研究グループでは通信のほとんどが事前に DNS による名前解決を行う点に着目し，DNS にクライアント IP アドレスを通知する機構を組み込むことにより，問合せ元に応じて動的にファイアウォールの検査内容を決定するシステムを提案しているが，現在具体的な実装方法は示されていない．そこで本研究では，DNS 拡張機能（EDNS0）によりクライアントのサブネットアドレスとネットマスクを DNS 問合せに埋め込む機能を用いて DNS キャッシュサーバによるクライアント IP アドレス通知機能の実装を行った．

Design and Implementation of Client IP Address Notification Function on DNS for Proactive Firewall System

Tomokazu Otsuka[1]　Gada[1]　Nariyoshi Yamai[2]　Kiyohiko Okayama[3]

1. まえがき

今日，ファイアウォール製品や UTM 製品は不正アクセスに対する一般的な対策として導入されており，組織内と組織外との通信を検査している．しかし多くのファイアウォールでは負荷の高い処理を行うとスループットの低下を招くため，これを回避するために監視対象となる通信を限定したり，負荷の高い検査を行わないようにしたりするなどの設定を行う必要がある． さらに，このような構成のほとんどは管理者が手動で行わなければならないため，管理者の負担が大きくなるという問題がある．この問題に対して，本研究グループでは TCP/IP 通信で原則として事前に DNS[1][2] による名前解決が行われている点に注目し，クライアント側の DNS サーバがクライアント IP アドレスをファイアウォールに通知する機能を提案する．たとえば送信元が信頼できる場合にはファイアウォールをバイパスにしたり，負荷の高い検査を行わないようにしたりして高速通信を許可する一方，通信相手が信頼できない場合には帯域を制限したり，負荷の高い検査を行ったりすることが可能になる．また，ボットを発信源とする通信の多くに見られるような，名前解決を行わないような通信については，不正アクセスとみなにて遮断することも可能である．これにより，信頼できる通信と疑わしい通信とを分離し，信頼できる通信の高速化を図るとともに管理者の省力化にもつながる．そこで本稿では，提案されているシステムの一部である，クライアント側 DNS サーバがクライアント IP アドレスをサーバ側の DNS に通知する機能の設計と実装について報告する．

以下，2 章では従来のファイアウォールの問題点と提案システムの実現方針を述べ，3 章で DNS キャッシュサーバの設計について述べ，4 章では提案システムの実装と動作結果を述べる．最後に，5 章で本論文のまとめ，今後の

[1] 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University
[2] 東京農工大学大学院工学研究院
Institute of Engineering, Tokyo University of Agriculture and Technology
[3] 岡山大学情報統括センター
Center for Information Technology and anagement, Okayama University

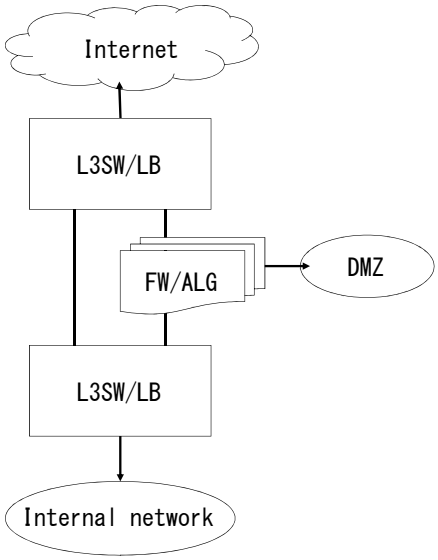

図 1　想定するネットワーク環境
Fig. 1　Assumed network environment

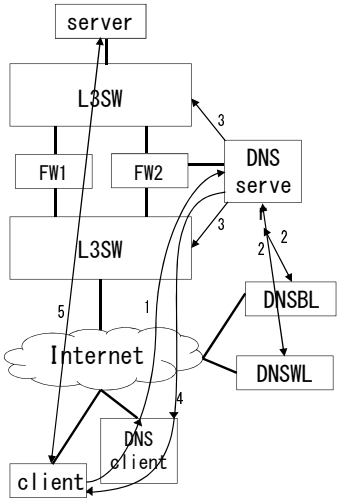

図 2　システム全体の構成
Fig. 2　An example of system structure

課題について述べる．

2. DNS との連携による動的ファイアウォールシステム

2.1 ネットワーク環境

本研究で想定しているネットワーク環境を図 1 に示す．この図において，レイヤ 3 スイッチ（L3SW）／負荷分散装置（LB）は送信元や送信先の IP アドレスやポート番号等のレイヤ 3，4 の情報に基づいて特定の条件を満たすフローをファイアウォール（FW）やアプリケーションゲートウェイ（ALG）に振り分け迂回させる機能やパケットを目的の IP アドレスに対応する出力ポートに転送する機能を持つ．FW は主にレイヤ 4 より上位のレイヤの情報に基づいて検査を行う．ALG はネットワークに出入りするアクセスの管理をし，HTTP や SMTP などの特別なアプリケーションに関して，たとえば電子メールにおけるウィルス検査など FW で実施するのが困難，あるいは適切でないような検査を行う．SSL-VPN 装置なども便宜上 ALG に含めるものとする．FW や ALG はそれぞれ複数台あってもよい．また，1 台の装置で仮想的に複数の FW や ALG の役割を果たしてもよい．このようなネットワーク環境は多くの組織に見られるもので，またそれ以外の組織でも比較的容易にこのようなネットワーク環境を構成可能であると考えられる．

2.2 従来のファイアウォールの問題点

ファイアウォールは，組織外のコンピュータネットワークから組織内ネットワークへの侵入を防ぐシステムであり，外部のネットワークから侵入・データ改ざん・破壊などの攻撃を受けないように外部と内部との境界の通信を監視し不正アクセスやさまざまな攻撃から守っている．
一般にスループットは検査内容によって大きく異なり，SPI（Stateful Packet Inspection）[3] やウィルス検査などの負荷の大きな処理を行うとスループットが低下する．また，最大同時接続セッション数や最大新規接続セッション数にも制限があり，組織ネットワークの規模とファイアウォールの性能によっては一部の通信が行うことができなかったり遅延が発生したりするという問題がある．

この問題に対して，たとえばブラックリスト（BL）[4] やホワイトリスト（WL）[4] のように，安全性の高い一部の通信に関してはファイアウォールを迂回させるようにレイヤ 3 スイッチ/負荷分散装置の設定を行ったり，負荷の高い検査を省略したりするようにファイアウォールを設定するようにする．一方，危険性の高い一部の通信に関してはレイヤ 3，4 の情報に基づいてレイヤ 3 スイッチ/負荷分散装置で遮断する方法がある．ただし，この方法では管理者が手動で設定を行う必要があるため，ファイアウォールの設定の増加に比例して管理者の負担が増大するという問題もある．

2.3 動的ファイアウォールシステムの概要と方針

2.2 節で述べた問題点を軽減する方法として，送信元・送信先情報に基づいてレイヤ 3 スイッチ/負荷分散装置やファイアウォールの設定を動的に変更するようなファイアウォールシステムが提案されている．以下システム全体の概要と基本方針を述べる．

2.3.1 動的ファイアウォールの概要

我々の研究チームでは外部で提供されている DNS ベースのホワイトリスト（DNSWL）及びブラックリスト（DNSBL）に基づいて動的にファイアウォールが検査内容を変更するシステムを考えた．[5] このシステム全体の構成を図 2 に示す．この図において，FW1 はホワイトリストに含まれる相手との通信用で，負荷の高い検査を省略したファイア

ウォールであり，FW2 はそれ以外の相手と通信するときに用いられるファイアウォールを表している．ただし，初期状態のすべての通信は FW2 を通るようにレイヤ 3 スイッチ（L3SW）で設定されているものとする．この構成図において，インターネット上のクライアントが組織内ネットワーク上のサーバにアクセスする場合の動作手順を以下に示す．ただし，作業中の番号は図中の番号と対応している．

(1) クライアントはクライアント側 DNS サーバ（DNSclient）にサーバの名前解決を依頼する．クライアント側 DNS サーバはクライアントの IP アドレスを含めた問合せパケットをサーバ側 DNS サーバ（DNSserver）に送信する．

(2) サーバ側 DNS サーバは問合せメッセージ中にクライアントの IP アドレスが含まれていればそれを取り出し，DNSBL，DNSWL に登録されているかどうかを確認する．クライアントの IP アドレスが含まれていなければ，(4) に進む．

(3) サーバ側 DNS サーバは DNSWL，DNSBL への登録の有無によりレイヤ 3 スイッチの設定を変更する. すなわち，DNSWL に登録されている場合はクライアントとサーバとの通信を FW1 経由で行うように制御する.DNSWL に登録されておらず DNSBL に登録されている場合は，クライアントの IP アドレスを送信元あるいは送信先として含むパケットを破棄するようにレイヤ 3 スイッチを変更する. クライアントの IP アドレスが DNSWL，DNSBL のいずれにも登録されていない場合や問合せメッセージ中にクライアントの IP アドレスが含まれていない場合はレイヤ 3 スイッチの設定は特に変更しない．

(4) サーバ側 DNS サーバはクライアント側 DNS サーバにサーバの IP アドレスを含む応答を送信する. クライアントの IP アドレスが DNSBL に登録されている場合，(3) でレイヤ 3 スイッチの設定は変更せず，サーバの IP アドレスの代わりに他の IP アドレス（ハニーポットやレイヤ 3 スイッチで事前に設定しておいたパケット廃棄用 IP アドレス）を応答してもよい. クライアント DNS サーバはサーバの IP アドレスを得てクライアントに通知する．

(5) クライアントはサーバとの通信を開始する．

2.3.2 システムの実現方針

TCP/IP では送信元・送信先の識別子として IP アドレスを用いて通信を行っている．しかし，利用者が直接 IP アドレスを指定することは稀であり，通常は可読性に優れているホスト名（ドメイン名）を用いる．その際，名前解決（ホスト名から IP アドレスに変換）に用いられるのが DNS である．

現在の DNS プロトコルでは名前解決を行いたいクライアントの情報は問合せメッセージに含まれておらず，またキャッシュサーバの存在で通信する度に必ず問合せを行うとは限らない．しかし，アプリケーションプロトコルによる通信を行う前に問合せが発生するという性質は重要な特徴である．そこで，この性質に注目し，DNS を送信元（クライアント）の IP アドレスを通知する機能を持つように拡張することで問合せ先（ファイアウォール側）が送信元・送信先情報を事前に把握でき，ファイアウォールシステムの設定を動的で変更することを可能とする．

そこで本研究では，DNSclient によってクライアント IP アドレスを DNSserver に通知する機能の設計と実装を行う．異なるクライアントから同じホスト名に対する問合せが発生すると，従来の DNS ではキャッシュ機能によりクライアントの IP アドレスが通知されない場合があるという問題が生じるので, 同一のホスト名に対する問合せであってもクライアントが異なる場合にはキャッシュ機能を無効化する方法も提案する．

次章で，この方針を実現するためのシステム設計を述べる．

3. DNS キャッシュサーバの設計

3.1 問合せ元 IP アドレスの通知

提案システムでは，既存の DNS プロトコルと互換性を保ちながら問合せ元の IP アドレスをサーバ側 DNS サーバに通知する機能が必要となる. 通知方法としては，EDNS0[6] を活用するなどして DNS の問合せメッセージに問合せ元（クライアント）の IP アドレスを含める方法などが検討されているが，現在ではインターネットドラフトとなっているクライアントサブネットオプション [7] を利用してクライアント IP アドレスを通知する方法 [8] が提案されている. この方法は EDNS0 を利用してクライアント側 DNS サーバがクライアントのサブネットアドレスとネットマスクの組（以下，クライアントサブネットオプションとする）を問合せに埋め込むものである. 基本的にはネットワーク単位での通知を想定していると考えるが，ネットマスクを 32 ビットとすることでクライアントの IP アドレスを通知することも可能である. ところが次のような問題が生じる．

3.2 キャッシュ機能の一部無効化

DNS では問合せ回数を軽減するためにキャッシュ機能が設けられている. 各資源レコードに対して指定された有効期限（TTL: Time to Live）の間は同一資源レコードの問合せは行わないようになっている. しかし，提案方式ではこの機能により問合せ元 IP アドレスの通知が行えない場合が生じる. すなわち，同一 DNS サーバを利用するクライアントのうち 1 つがサーバにアクセスしようとして名前解決を行うと，この（クライアント側）DNS サーバではサーバの IP アドレスがキャッシュされるため，キャッシュの有効期限内に他のクライアントが同一サーバにアクセスし

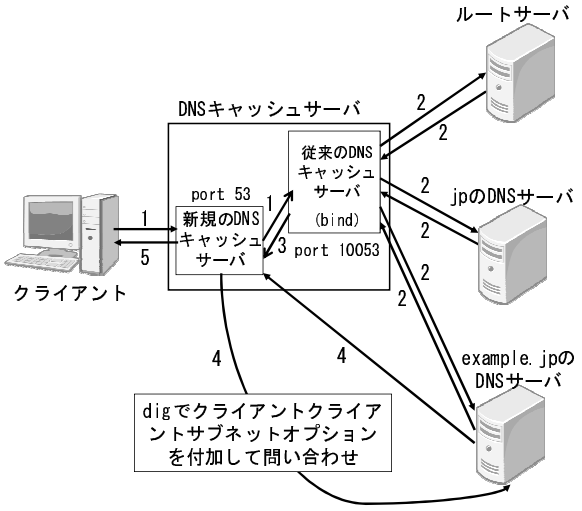

図 3　システムの構成例

Fig. 3　Configuration example of the proposed system

てもそのクライアントの IP アドレスはサーバ側に通知されない.

この問題を解決するためには，キャッシュ機能の一部無効化が必要となる. すなわち異なるクライアントからの問合せを受けた場合，クライアント側 DNS サーバは以前の問合せ結果（名前解決をしたサーバの IP アドレス）を含む資源レコード（A レコード，AAAA レコード）のキャッシュを無視し，再度同じ名前解決を行うようにする．ただし，クライアントの IP アドレスの通知はサーバ側 DNS に対して 1 回だけでよいため，NS レコード，MX レコードなど，サーバの IP アドレスを含まない資源レコードについてはキャッシュを無効化する必要はない.

4. システムの実装と動作確認実験

本章では，3 章で示したシステムの実装と動作確認について示す.

4.1 DNS キャッシュサーバの実装

システムの構成例として図 3 に示す．問合せ元 IP アドレスの通知とキャッシュ機能の一部無効化を実現するために，新規の DNS キャッシュサーバ（以下，新規キャッシュサーバとする）を 53 番ポートで実装し，従来の DNS（bind）を 10053 番ポートで受けるように変更した．新規キャッシュサーバの構築にあたり以下の手順で DNS キャッシュサーバを実装した.

4.1.1 クライアント IP アドレスの通知機能

新規キャッシュサーバには CPAN の Perl モジュールの Net::DNSServer::Proxy [9] を利用した．このモジュールは，クライアントから問合せを受けると指定した他の DNS（本研究では 10053 番ポートの従来の DNS）に問合せを受け流すものである．例として www.example.jp の名前解決を以下で述べる.

(1) 新規の DNS キャッシュサーバは bind にクライアントからの問い合わせを受け流す.
(2) bind は新規の DNS キャッシュサーバからの問い合わせを元に反復問い合わせにより名前解決を行う.
(3) 新規の DNS キャッシュサーバは bind から応答を受ける.
(4) 新規の DNS キャッシュサーバは bind からの応答から最終的に応答してきた example.jp の DNS サーバの情報を取り出し，この DNS サーバにクライアントサブネットオプションを付加して再度問い合わせを行う．また，このときの問い合わせでは bind に付属されてある問い合わせ用プログラムである dig を用いる.
(5) クライアントに応答するのは bind からの結果を応答するようにする.

本研究では，bind の反復問い合わせの際にクライアントサブネットオプションを付加するのではなく，最終的に応答してきた example.jp の DNS サーバにのみ dig によりクライアントサブネットオプションを付加した問い合わせを行う．その理由としてセキュリティ対策が挙げられる．クライアントサブネットオプションは反復問い合わせの中に含めることができるが，反復問い合わせを行う過程で関係のない権威サーバにクライアントの IP アドレスを通知する必要はない [10]．また，関係のない権威サーバ付近でこのやり取りに関するパケットを盗聴される恐れがあり，そのリスクを下げるためである.

4.1.2 キャッシュの一部無効化機能

新規キャッシュサーバに 3.2.2 節で述べたキャッシュの一部無効化する機能を設けるために，以下のようにキャッシュを一部無効化するか否かを判断する．あるクライアントからの問合せが発生すると bind に問合せを渡す．bind は名前解決を行ったのちキャッシュサーバに応答すると同時にクライアント IP アドレス・問合せ先 FQDN・無効時刻をキャッシュしておく．ここで無効時刻とは，1970 年 1 月 1 日の 00:00:00 から現在までの経過時間と権威サーバが返してきた A レコードに含まれている TTL との和で表された値で，新規キャッシュサーバ内でキャッシュが有効な時間を意味する．また異なるクライアントからの問合せが発生すると bind に渡すまでは同様だが，bind は自身でキャッシュを保持しているため名前解決を行うことなくキャッシュサーバにそのまま応答する．しかしクライアント IP アドレスとそれに対応する FQDN はキャッシュになく，キャッシュにヒットしないため再度 dig により問合せを行う．新規キャッシュサーバに応答すると同時にクライアント IP アドレス・問合せ先 FQDN・無効時刻をキャッシュしておく．キャッシュにヒットする場合は dig による問合せを行わず bind からの応答をクライアントに返す．クライアント IP アドレスと問い合わせ先 FQDN のセット

```
; <<>> DiG 9.9.4 <<>> @ns1.google.com IN A +client=150.46.47.121
; (1 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35717
;; flags: qr aa rd ; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0,flags:; udp: 512
; CLIENT-SUBNET: 150.46.47.121/32/22
;; QUESTION SECTION:
;www.google.com              IN    A

;; ANSWER SECTION:
www.google.com      300   IN    A    173.194.38.52
www.google.com      300   IN    A    173.194.38.48
www.google.com      300   IN    A    173.194.38.50
www.google.com      300   IN    A    173.194.38.49
www.google.com      300   IN    A    173.194.38.51

;; Query time: 64 msec
;; SERVER: 216.239.32.10#53(216.239.32.10)
;; WHEN: Fri Jan 17 13:00:13 JST 2014
;; MSG SIZE  rcvd: 135
```

図 4　キャッシュサーバでの応答結果
Fig. 4　Response results in the cache server

をキャッシュしておくことで，クライアント IP アドレスを含めてキャッシュにヒットしているかを調べることができる．

4.2　動作確認実験

4.2.1　実験環境

本研究で実装したシステムの実行環境は表 1 に示す．

表 1　実行環境

OS	FreeBSD/amd32 8.2-RELEASE
割当 CPU	Intel(R) Xeon(R) 2.40GHz
割当メモリ	1.0GB
割当 HDD	32GB

4.2.2　動作結果

実験方法としては，クライアントサブネットオプションをサポートしている Google のパブリック DNS にアクセスした．その結果として DNS キャッシュサーバでの問い合わせ及び応答の一例を図 4 に示す．その結果想定通り最終的に応答してきた Google の DNS に自身の IP アドレスを付加して送れているのが確認できる．またコンテンツサーバ側でクライアント IP アドレスを認識しているのが確認できた．以上より期待していた結果となった．

5.　むすび

本論文では，DNS を用いたクライアント IP アドレスの通知機能システムを提案し，実装および動作確認を行った．従来の DNS では，名前解決を行いたいクライアントの情報は問合せメッセージに含まれていなかったためファイアウォールが送信元・送信先情報を事前に把握し動的な検査を行うことができなかった．しかし今回の研究で，ファイアウォールが動的に検査内容を決定するためのシステムの一つとして，DNS を用いて送信先にクライアント IP アドレスを通知することが可能となった．また本システムは問合せ元 IP アドレスの通知機能を前提としているが，クライアント側 DNS サーバが未対応の場合でもある程度有効である．たとえば電子メールにおける SPF（Sender Policy Framework）と同様にクライアント側 DNS サーバが問合せを受ける IP アドレスの範囲を公開するような仕組みを導入すれば，問合せ元 IP アドレスの特定はできないが，その代わりに問合せ元 IP アドレス範囲を絞り込むことが可能になり，その結果に応じてファイアウォールの動作を変更できると考える．今後の課題として，ファイアウォールが動的に検査内容を決定できるように本システムを用いて問合せメッセージからクライアント IP アドレスを取り出すことで DNSWL，DNSBL に登録されているか確認し，レイヤ 3 スイッチと連携するようにサーバ側 DNS サーバを構築するとともに，その有用性を検証していく必要がある．

謝辞

本研究の一部は平成 25〜27 年度科学研究費補助金 (基盤研究 (C)，課題番号 25330105) の補助を受けている．ここに記して感謝の意を表する．

参考文献

[1] P.V. Mockapetris: Domain Names - Concepts and Facilities, RFC1034, IETF, 1987.
http://www.ietf.org/rfc/rfc1034.txt

[2] P.V. Mockapetris: Domain Names - Implementation and Specification, RFC1035, IETF, 1987.
http://www.ietf.org/rfc/rfc1035.txt

[3] Seungyong Yoon, Byoungkoo Kim, Jintae Oh and Jongsoo Jang: High Performance Session State Management Scheme for Stateful Packet Inspection,Managing Next Generation Networks and Services, Lecture Notes in Computer Science,Vol.4773,pp.591-594,2007.

[4] J.Levine: DNS Blacklists and Whitelists, RFC5782, 2010.
http://tools.ietf.org/search/rfc5782

[5] 岡山聖彦，山井成良，ガーダ，大塚友和: DNS と OpenFlow スイッチとの連携による動的ファイアウォール，インターネットと運用技術シンポジウム 2013（IOTS2013）論文集，pp95-98，2013

[6] P.Vixie: Extension Mechanisms for DNS(EDNS0), RFC2671, IETF,1999.
http://www.ietf.org/rfc/rfc2671.txt

[7] Index of /edns-client-subnet, 2013.
http://wilmer.gaa.st/edns-client-subnet/

[8] C.Contavalli，W.van der Gaast, S.LEACH and E.Lewis: Client Subnet in Draft, Work in progress, IETF, 2013.

http://tools.ietf.org/html/draft-vandergaast-edns-client-subnet-02
[9] R.Brown: Net::DNSServer::Proxy, 2002. http://search.cpan.org/~bbb/Net-DNSServer-0.11/lib/Net/DNSServer/Proxy.pm
[10] J.Damas, M.Graff, P.Vixie: Extension Mechanisms for DNS (EDNS(0)), RFC6891, 2013. http://tools.ietf.org/search/rfc6891