挙動に基づくDNSアンプ攻撃の検知

蔡, 龍洙
九州大学大学院システム情報科学府

馮, 尭鍇
九州大学大学院システム情報科学研究院 : 助教

川本, 淳平
九州大学大学院システム情報科学研究院

櫻井, 幸一
九州大学大学院システム情報科学研究院

出版情報：コンピュータセキュリティシンポジウム. 2015, pp.2F4-4-, 2015-10-21
バージョン：
権利関係：

挙動に基づく DNS アンプ攻撃の検知

蔡 龍洙† フォン ヤオカイ†† 川本 淳平†† 櫻井 幸一††

†九州大学大学院システム情報科学府

†† 九州大学大学院システム情報科学研究院
819-0395 福岡市西区元岡 744 番地
cailongzhu@itslab.inf.kyushu-u.ac.jp,

fengyk@ait.kyushu-u.ac.jp, {kawamoto,sakurai}@inf.kyushu-u.ac.jp

あらまし 近年, DNSサーバを利用したDNSアンプ攻撃が増加している. DNSアンプ攻撃は, DNSサーバへの問い合わせ及びキャッシュ機能を利用してデータの量を増幅させる．そのデータを攻撃対象とするサーバへ送信することで，回線をパンクさせる．コントロールされた多数のコンピュータから一斉に大量のデータを送りつけて対象を麻痺させるDDOS攻撃の一種である．小さなDNS要求パケットを送るだけで，その何十倍に達する大量の応答が生成され，被害者側のネットワークが混雑させられ，通常サービスがダウンしてしまう可能性もある．DNSアンプ攻撃を検出するには既にさまざまな方法が提案されたが，パラメーターのチューニングが必要であり，そのパラメーターは簡単に決められない場合が多い．本研究ではDNSアンプ攻撃を挙動に基づいて検知する方法を提案し，その提案を実験で検証する．

A Behavior-based Proposal for Detecting DNS Amplification Attacks

Longzhu Cai† Yaokai Feng† Junpei Kawamoto† Kouichi Sakurai†

†Kyushu University
744 Motooka, Nishi-ku, Fukuoka 819-0395, JAPAN
cailongzhu@itslab.inf.kyushu-u.ac.jp,
fengyk@ait.kyushu-u.ac.jp,{kawamoto,sakurai}@inf.kyushu-u.ac.jp

Abstract DNS amplification attack has become a popular form of the attacks of the Distributed Denial of Service (DDoS) in recent years. In DNS amplification attacks, the attackers utilize spoofed source IP addresses and open recursive DNS servers to perform the bandwidth consumption attacks. A large amount of responses are generated and they are sent to the targets after the attackers send only a little of DNS requests. Various methods have been proposed for detecting the DNS amplification attacks. However,

almost of them have to determine parameters in advance, which is not easy for many cases. In this study, we propose a behavior-based method for detecting DNS amplification attacks and its performance is verified by experiments.

1 はじめに

2013 年 3 月に英国の非営利スパム対策組織「Spamhaus.org」を対象とした DDOS 攻撃が行われた．「Spamhaus.org」という団体は，非常に多くのインターネットユーザのスパムフィルタリングを支援している．この攻撃で，「Spamhaus.org」組織のサーバがダウンしてしまい，ユーザの受信トレイが大量のスパムで溢れかえることになった．その時，利用された攻撃の方法が DNS アンプ攻撃だった. DNS アンプ攻撃は一つのネット資源を消耗する攻撃の方式である．現在，ネット上の多くのオープンリゾルバ群が「増幅器」として攻撃者に頻繁に利用されている．

現在，多くの DNS サーバは EDNS（Extension Mechanisms for DNS）機能を持っている．この仕組みにより DNS の応答が 512 バイトを超える時にも UDP で送信することができる．本来はユーザに便益をもたらすために作られた機能であるが, DNS アンプ攻撃では攻撃者に悪用されることになった. DNS アンプ攻撃に関する既存研究では, さまざまな検知方法と対策が提案されている．しかし，攻撃を検知する際，検知方法自体がサーバに負荷をかけたり，パラメーターを多く導入したいるため検知の汎用性が低いという問題がある．すなわち，各 DNS サーバの状況と性能には差があることが普通であり，ある環境に対して最適化された手法が他の環境において同等の効果を発揮できない恐れがある．我々は各 DNS サーバに対して，それぞれの過去データから学習したパターンを用い, 新たな攻撃が発生した際に攻撃を検出する方法を提案する．

我々が提案した手法は DNS アンプ攻撃が発生している際に, DNS サーバ側から検知できるリクエストの頻度と増幅されたデータ量の比率を二次元の特徴ベクトルとして表現する．更に，パターン認識の手法を用いて過去データにより生成されたパターンを基に正常か異常かを判断する．

2 DNS の仕組み

DNS プロトコルは基本的に UDP（User Datagram Protocol）を使っている．UDP はコネクションレス型プロトコルで転送効率が高く，遅延が小さいため，高速性や即時性を重視する用途でよく利用されている. しかし，確実性を重視する TCP（Transmission Control Protocol）と比べて信頼性が低いことが指摘されている．DNS アンプ攻撃では，攻撃者はこのようなプロトコルの欠点を利用し，攻撃を行う．

DNS 検索には再帰（Recursive）検索と反復（Iterative）検索との二つの方式がある[1]. 先ず，この二種類について説明する．

2.1 再帰検索

「dns.example.com」の名前解決を例として説明する（図 1 を参照）．

① クライアント側からローカルサーバに名前解決要求を行う．
② ローカルサーバはキャッシュから関連する記録を探す．存在しなければルート DNS サーバに送信する．
③ ルートサーバはリクエストを受けた後, ローカルサーバに「.com」ドメイン名に対応するトップ DNS サーバのアドレスを送信する．
④ ローカルサーバはまた「.com」トップ DNS サーバに名前解決要求を送信する．
⑤ 「.com」トップサーバはキャッシュから対応する記録を探す．存在していれ

ば応答し，存在していなければローカルサーバに「.example.com」ドメイン名に対応するサーバのアドレスを送信する．

⑥ 同様な操作を繰り返す．

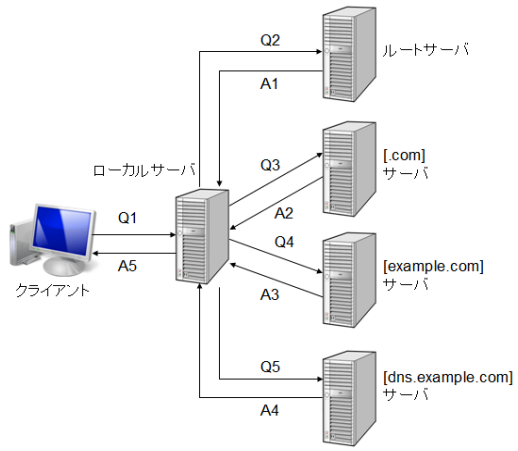

図１ 再帰検索の例[2]

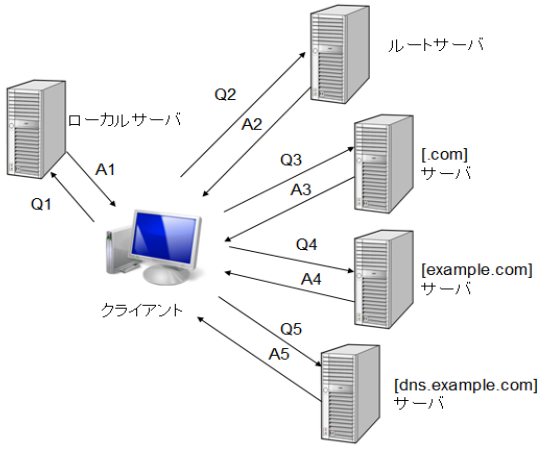

図２ 反復検索の例[2]

2.2 反復検索

2.1 節と同様に「dns.example.com」の名前解決を例として説明する(図２を参照)．

① クライアント側からローカルサーバに名前解決要求を行う．

② ローカルサーバはキャッシュから対応する記録を探す．あったら応答し，なかったらクライアントにルート DNS サーバのアドレス情報と DNS 応答パケットを送信する．

③ クライアントは応答を受けた後，ルート DNS サーバにリクエストを送信する．

④ ルート DNS サーバはリクエストを受けた後，クライアントに「.com」ドメイン名に対応するトップ DNS サーバのアドレス情報と DNS 応答パケットを送信する．

⑤ 同様な操作を繰り返す．

DNS アンプ攻撃は準備段階で DNS サーバの反復検索機能を利用する．

3 DNS アンプ攻撃

DNS アンプ攻撃とは，DNS サーバのキャッシュ機能を悪用する DDOS 攻撃である[3]．

3.1 DNS 攻撃の流れ

一般に DNS サーバと呼ばれるコンピュータもしくはサービスには，大きく分けて二つの機能がある．DNS ゾーン情報を外部に対して提供する「権威サーバ」と，クライアントからの名前解決要求を処理する「キャッシュサーバ」がある．図４で，乗っ取られた DNS サーバが権威サーバで，オープンリゾルバサーバ群がキャッシュサーバである．

ステップ１

攻撃者はウィルスなどを使い，ネット上の脆弱性がある権威サーバに侵入する．次に，用意した長いファイルを乗っ取った DNS サーバに登録する．図３にステップ１の概要を示す．

ステップ２

① 攻撃者は事前に遠隔管理ソフトを仕込んだゾンビ PC を利用し，キャッシュサーバに一斉に該当レコードの問合せをさせる．（ゾンビ PC は攻撃者に侵入されたネット上の脆弱性があるコンピュータを指す．）

② ゾンビPCは乗っ取られたDNSサーバから攻撃者により登録されたレコードファイルを取得する．

③ ゾンビ PC は，取得したレコードをキャッシュする．

④ 攻撃者は被害者 IP アドレスを自分の IP アドレスに偽装し，該当レコードを問い合わせる．

⑤ キャッシュサーバから大きなサイズのレコード情報が被害者側に送信される．被害者側のネットワークが混雑し，ダウンしてしまう．

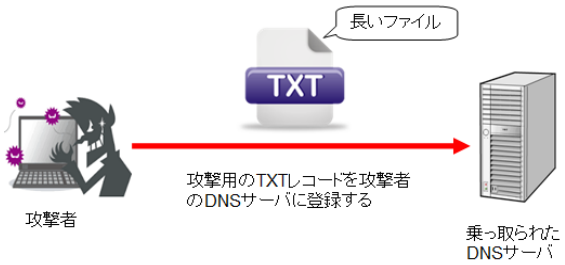

図３　ステップ１

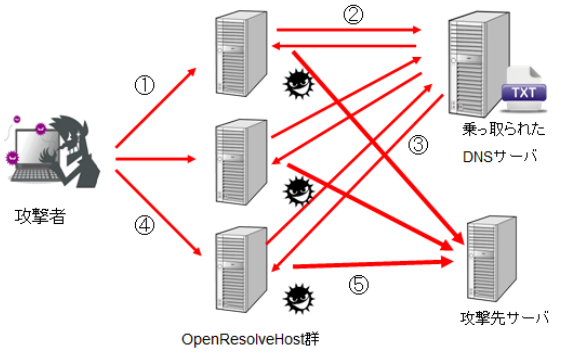

図４　ステップ２

4　既存研究

DNS アンプ攻撃を検出･制限するにはさまざまな方法が提案されている．サーバが記録した応答レートによる検知[4]，リクエストと応答の対応関係を表した表による検知[5]，偏差値によりサーバ群から異常サーバを検出する方法[6]などが挙げられる．

Vixie 氏と Schryver 氏により提案された DNS アンプ攻撃の制限する方法[4]では，DNS 問い合わせの頻度により検知することではなく，応答頻度を記録する．これが一定の閾値を超えた時に該当の問合せの受理をやめる．このような操作により応答頻度は下がる．応答頻度が閾値より下がればまた受理を再開する．すなわち，一つの問い合わせの応答頻度が閾値を超えないように確保する．従って，DNS アンプ攻撃の効果を低減することが実現できる．しかし，各 DNS サーバの閾値を決める必要がある．閾値の決め方によって正常の問合せが誤検知されずに攻撃の影響を最大限に下げることができるかが決まる．

リクエストと応答の応対関係を表した表による検知する方法[5]では，問合せと応答の情報により，DNS アンプ攻撃を検知することができる．しかし，この方法自体は大量の対応関係を記録することにより，サーバに負荷がかかる．場合によっては，逆に効率が低下する傾向がある．

Ma らは偏差値により DNS サーバ群から異常 DNS サーバを検出する方法[6]を提案した．この方法では，管理者が管理しているすべての DNS サーバの中で，同時に攻撃を受けるものは少数であるという仮定がある．この論文[6]では，各 DNS サーバを通したデータ量とリクエストの接続数などを記入し，平均値を求めた後，各 DNS サーバが平均値に対する偏差値を求め，この偏差値が一定の値を超えた時に異常の DNS サーバだと検出する方法である．しかし，異常な DNS サーバを検知するのに利用された閾値は作者が自分の経験により決めたものである．従って，他の DNS サーバ群に対してこの検知方法を適用する場合には閾値の決定が一つの問題点となる．

5　提案手法

本研究では過去のデータから時間単位ごとにリクエスト数と拡大倍数（応答データ量がリクエストデータ量に対する比率）という二

つの特徴量を抽出し，機械学習を用いて DNS アンプ攻撃の分布領域を識別する方法を提案する．本提案により閾値を決定する問題からパターン認識の問題に転換した．また，一つ一つの IP アドレスを記録する必要がないため，DNS サーバに大きな負荷をかけない．

5.1　提案の詳細

図 5 に示したように，X 軸は単位時間のリクエスト数で，Y 軸は単位時間の DNS サーバからの応答数のリクエスト数に対する比率である．DNS アンプ攻撃は主にレコードのサイズで拡大するため，普通の DDOS 攻撃より拡大倍数が大きい．普通の DDOS 攻撃はサーバのサービス提供不能を目標とするので，攻撃期間に大量なデータを送るのが普通である．しかし，DNS アンプ攻撃のターゲットは DNS サーバではなく，DNS サーバの送信先のクライアントなので DNS サーバの処理能力を確保する必要がある．従って，普通の DDOS 攻撃よりリクエスト量が少ない傾向がある．そのため，単位時間当たりのリクエスト数とリクエスト数に対する応答数の比率で分類することができる．

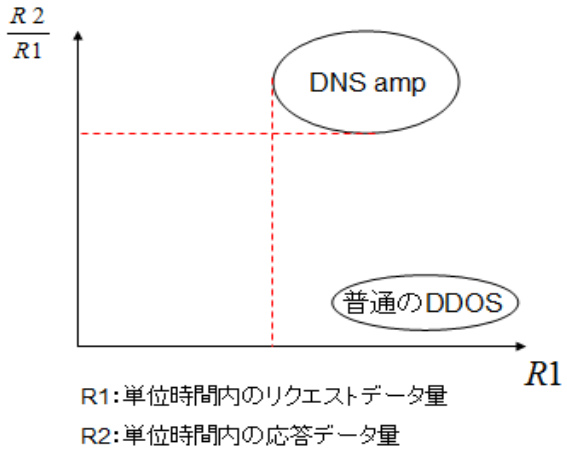

図 5　予備実験

次に，記録したデータを機械学習でグループに分ける．本論文では，k-means 法を用いて異常と正常の状況を分類する．

5.2　k-means アルゴリズム

k-means アルゴリズムは距離ベースのクラスタリングアルゴリズムである．すなわち，距離を類似度を判断する基準として利用し，距離が近い点を同じクラスターに集める．
まず，全部 N 個の中から任意の k 個のデータを選び，最初のクラスターの基準点とする．残りの点は k 個の基準点との距離を計算し，一番近い基準点のクラスターに入る．次に，各クラスター内で新しい基準点を計算する．また，すべての点から新しい基準点への距離を計算し自分の所属を再配置する．この様に何回も繰り返し，評価関数による収束条件を満たしたらアルゴリズムを終える．

アルゴリズムの流れ：

① 任意の k 個の基準点 $v_1, v_2, \ldots, v_k$ を選択する．
② 残りの N − k 個の点から各基準点への距離を計算し，近い基準点のクラスターに入れる．
③ 各クラスターの基準点を計算する．
④ 収束条件を満たすまで②～③の操作を繰り返す．

本論文ではユークリッド距離を k-means アルゴリズムに利用する．

$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$　　$\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

本実験で n は 2 を取る．x_1 は単位時間内に受けたリクエストの数（頻度）であり，x_2 は応答データ量とリクエストデータ量の比率である．すなわち，二次元の特徴ベクトルを利用し，距離を計算する．

6　実験

上述したように本論文の提案では，時間単位ごとに DNS サーバが受けたリクエストデータ量に対する応答データ量の拡大倍数とリク

エストの頻度に基づいて検知する. すなわち, 時間単位ごとにこの二つの特徴量を抽出して, k-means アルゴリズムを用いて正常と異常を分類する．本論文では，ユークリッド距離を利用する.

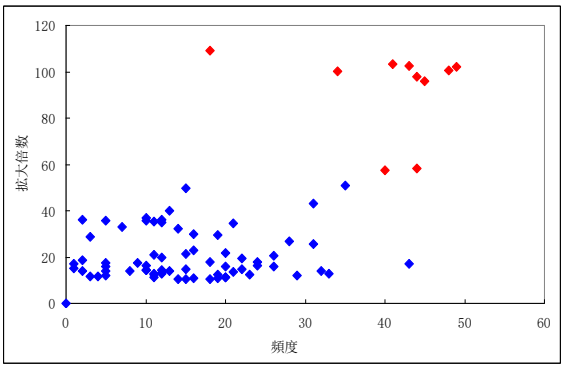

図 7　分類結果（k=2）

6.1　実験データ

実験データは Shumon Huque がブログで公開したデータ[7]を利用した. このデータセッ
トは Shumon Huque が事前に処理したもので，中にはリクエストごとにリクエストデータ量と応答データ量が記録されている．しかし，各パケットに Timestamp が付いていない. 我々はこのデータセット利用して DNS アンプ攻撃時の状況をアルゴリズムで模擬した.

（オリジナルデータの一部が付録 A に示されている）

6.2　実験データのプリプロセッシング

オリジナルデータセットの各パケットには頻度の情報がないので，我々は C 言語のランダム関数を利用して単位時間当たりのリクエスト数を模擬した．この実験では，0 から 49 までの値の中から任意の値を単位時間内のリクエスト数にした．しかし，ランダム関数だけ使ったら，分布が平均的な形になる可能性が高いと考えた．なので，実際の攻撃時の状況を考慮し，頻度が低い点の数（すなわち正常の点）を意図的に増えた．そうだとしても実際の攻撃の場合より正常の比率が低く，攻撃の点が比較的に多い設定になっているため，誤検知や曖昧な点が出られる可能性があると考えられる．このような設定で分類した結果が図 7，図 8 と図 9 に示されている.

6.3　実験結果

図 7 と図 8 に表したように, 横軸は頻度（模擬した単位時間内のリクエストの頻度）であり，縦軸は拡大倍数（単位時間内の応答データ量とリクエストデータ量の比率）である.

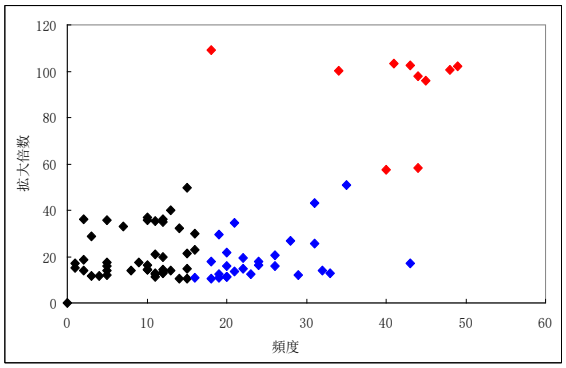

図 8　分類結果（k=3）

（1） k=2 の時

図 7 に示したように，全部の点は赤と青の二つのクラスターに分けられた．赤は攻撃と識別された部分であり，青は正常だと識別された部分である．正常か異常かを判断するのは人ではなく，プログラムで基準点の X 値と Y 値と比較することで容易に判断できる

ほかに，赤のクラスターの一番左のところに（18,109.17）という点が DNS アンプ攻撃クラスターに含まれた．この点は拡大倍数は高いが，頻度が低いため，正常の状態の可能性が極めて高い．しかし，このような点が誤検知され, DNS アンプ攻撃クラスターに入れられた原因はデータセットの欠陥だと考えられる．実際に攻撃が行った場合は，横軸の頻度の値が 5 倍（10 から 50 に）だけになることではなく，何十倍にもなることが普通である．その場合は，低頻度地域にある拡大倍数が高い点と DNS アンプ攻撃クラスターとの

距離が遠いため，誤検知が大幅に落ちると予想できる．

(2) k=3 の時

図 7 は k=3 の時の結果である．図 7 に示したように，赤の点は DNS アンプ攻撃だと識別した部分である．青の点は DNS アンプ攻撃状態と正常の状態が接するところにある．

また，青の部分と頻度が低い正常の部分が接するところの分類もはっきりできなかった．しかし，このところの点は頻度も拡大倍数も高くないため，攻撃の検知結果には直接に影響を与えないと考えられる．ただし，青のようなクラスターの平均値を減らし，青全体のクラスター範囲が左に移され，右側の点がほかのクラスターに入れられる恐れがある．

赤の（18,109.17）は k=2 の時と同じく誤検知された．

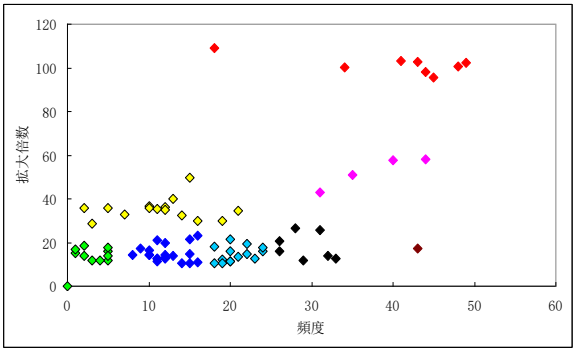

図 9　分類結果（k=8）

(3) k=8 の時

図 8 は k=8 の時の結果である．図に表したように，データは八つのクラスターに分けられた．赤の点は DNS アンプ攻撃状態を含んだクラスターであり，ピンクの点は拡大倍数が正常の状態とほぼ同じなので，DNS アンプ攻撃とは言えない．このような点が出たのはランダム関数により，実験を模擬した結果だと考えられる．実際の場合に攻撃者は目的を持ち，意図的に攻撃を行うため，このような攻撃らしい点は出ないと予想している．

k=8 の時と k=3 の時を比べると，クラスターの分類が明確であり，所属が曖昧な点が少なくなった．また，特殊な点（43,17.14）もほかの点から区別されることができた．

しかし，k=3 の時と同じく，（18,109.17）が誤検知になった．

7　まとめと今後の課題

本論文では，リクエストデータ量と応答データ量の関係における DNS アンプ攻撃を検知することを目標として k-means アルゴリズムを用いた分類方法を提案した．

提案手法では，既存の DNS リクエストと応答のデータ情報を処理し，DNS アンプ攻撃時の状況をアルゴリズムで模擬した．次に，k-means アルゴリズムを用いて正常の点と異常の点を分類しただけではなく，パラメータ k が実験結果に与える影響にも検討した．

今後の課題としては，時間の情報が付いているデータセットを手に入れ，実際の DNS アンプ攻撃時の状況を再現することである．また，結果図に出た（18,109.17）の誤検知の問題が実際に発生するかを調べる必要がある．もし発生しなかったら，誤検知の現象が出ないようにプログラムを改善し，模擬してから実験する．ほかに，k-means アルゴリズムだけではなく，ほかのクラスタリング手法を用いて結果を比較し，それぞれのメリットとデメリットを検討する．

謝辞

この研究の一部は，科学研究費（基盤研究（C）No. 25330131）の支援を受けている。ここに記して謝意を表す．

参考文献

[1] http://www.atmarkit.co.jp/ait/articles/0112/18/news001.html（アクセス日：2015/08/01）

[2] http://blog.csdn.net/lycb_gz/article/details/11720247（アクセス日：2015/08/01）

[3] http://e-words.jp/w/DNS_amp.html
（アクセス日：2015/08/01）
[4] Vixie P, Schryver V. “Dns response rate limiting(DNSRRL),”ISC-TN-2012-1-Draft1(2012).
[5] GEORGIOS K, TASSOS M, DIMITRIS G, et al. A fair solution to DNS amplification attacks[A]. Proceedings – 2nd International Annual Workshop on Digital Forensics and Incident Analysis[C].2007.38-47
[6] MA Yun-long, JIANG Cai-ping, ZHANG Qian-li, WANG Ji-long. “DNS abnormal behavior detection based on IPFIX”, (Information Technology Center, Tsinghua University, Beijing 100084,China)
[7] http://blog.huque.com/2013/04/dns-amplification-attacks.html
（アクセス日：2015/07/20）

付録 A　オリジナルデータ（一部）

query	response	amp
36	4085	113.47
36	4085	113.47
37	4093	110.62
36	4016	111.56
36	4016	111.56
37	4045	109.32
37	4045	109.32
38	4095	107.76
38	4095	107.76
38	4095	107.76
38	4095	107.76
38	4093	107.71
38	4093	107.71
38	4093	107.71
36	3972	110.33
36	3972	110.33
36	3965	110.14
36	3965	110.14
37	3979	107.54
37	3979	107.54