第 XVI 部

DNS extension and operation environment

第 16 部
DNS extension and operation environment

第 1 章　DNS ワーキンググループ 2006 年度の活動報告

DNS ワーキンググループでは、DNS に対する攻撃の調査と、それを防ぐための手段の研究や、DNSSEC もしくはそれに取って代わる新たなプロトコルの考案や、運用と普及に対する障害を解決するための研究、DNS の運用もしくは実装上発生する問題点の調査と、その解決方法の研究および議論を行っている。

今年度は、DNS に対する攻撃である cache poisoning のさまざまなやり方、およびその対処法などについて重点的に議論を行った。これらの議論をもとに、現在の主要な DNS 実装である BIND に対して攻撃への対策コードのパッチをいくつか開発し、公開を行った。

また、現在各地で行っている DNS のトラフィック解析、およびその結果についてのまとめと議論も行った。

以下、活動報告をまとめる。

第 2 章　DNS Response Size Issues

2.1 はじめに

本報告書は、DNS ワーキンググループのメンバが執筆した internet-draft である、draft-ietf-dnsop-respsize-06.txt について述べる。

2.2 draft-ietf-dnsop-respsize-06.txt の概要

draft-ietf-dnsop-respsize-06.txt で述べられているのは、DNS のパケットサイズに関する問題である。DNS のデータは主に UDP で送信される。そのため、RFC1035 の 4.2.1 にて 512 オクテット以下に収まるようにと決められている。しかし、これは IPv4 の UDP パケットサイズを想定して定義された値であり、IPv6 の最小 MTU は 1280 であるため、もっと大きなデータを転送することができる。

そのため、RFC2671 において、EDNS0 拡張が提案されている。EDNS0 では、512 オクテットを超えるサイズのデータを転送することができる。しかし、既存のリゾルバ実装等では、データサイズの最大値は依然として 512 オクテットであり、EDNS0 が普及するまでは 512 オクテットの限界というものは存在する。

そこでこの internet-draft では、既存の実装と EDNS0 実装が混在する移行期において、ゾーンの委譲の際に必要な情報が分断されてしまわないよう、ゾーン管理者ならびに DNS サーバ実装者へのアドバイスとして明記している。

2.2.1 ゾーン管理者へのアドバイス

ゾーン管理者は、問い合わせに対する応答がどの程度の大きさになるか、あらかじめ計算することが可能である。また、そのための Perl スクリプトもこの internet-draft に付属しており、例として Root DNS の NS レコード応答のメッセージサイズを分析して、如何に 512 オクテットに収めるかを述べている。そして、やはり最良なのは、問い合わせに対する応答が分断されないサイズに収まることであると述べている。

2.2.2 サーバ実装者へのアドバイス

DNS サーバを実装する際に、どの順番で応答データをつめていけば、512 オクテットを超えて途中で分断された場合にも必要な情報が欠落しないか、という観点からアドバイスが書かれている。NS レコードを応答する場合にも、ひとつの NS レコードに対して、そのグループとなる A と AAAA レコードを付加してから次の NS レコードを返す、といったアドバイスである。

また、NS レコードを並べた際に、512 オクテットに入りきる理論限界の値を示すことで、効率的な NS レコードの並べ方を示している。

2.3 おわりに

現在の GTLD-SERVERS.NET や ROOT-SERVERS.NET の場合には、NS レコード応答の際にすべてに同じ名前が含まれているため、メッセージを効率的につめることができている。このような現状の経験則をもとに、この internet-draft では、512 オクテットに収めるためのアドバイスが述べられている。

第 3 章　DNS フルリゾルバでの ID 詐称攻撃の検知

3.1 DNS フルリゾルバへのキャッシュ汚染攻撃

DNS への攻撃については、RFC 3833 にまとめられている。最近、そのうちの ID Guessing and Query Prediction 型の攻撃（あるいは Birthday attack）について話題にのぼっているが、現実に攻撃が行われているかは検証できていない。そこで、ID Guessing 型の攻撃を検出するしくみを作成することとした。そのうち、攻撃のパケットが多い場合は IDS などの別の方法で検出できるため、攻撃パケットが少ない場合を想定し、フルリゾルバで検出してログに出すこととし、BIND 8、9 のフルリゾルバへのパッチを作成した。さらに、現実に使用しているフルリゾルバにパッチを適用し、実環境での調査を行った。詳細を以下に述べる。

3.2 フルリゾルバでの検出方法

フルリゾルバは、反復検索の時に検索ごとにユニークな query ID（以下 QID）をつけ、権威サーバに問い合わせを行う。権威サーバから応答パケットを受け取ると、QID をキーとして問い合わせ処理を再開する。ID Guessing 型攻撃は、さまざまな QID を持つ応答パケットをフルリゾルバに送り、確率的に成功することを狙う攻撃方法であるため、フルリゾルバには送っていない QID の応答パケットが届くこととなる。QID が一致したとしても、DNS 応答パケットの内容を見て、問い合わせと異なる応答であれば捨てられる。

BIND 8、9 のリゾルバでは、DNS 応答パケットの QID を解釈し、送った検索に対応する応答か検索する部分がある。従来は、ログレベルを高く設定した場合に限り、情報量が少ないログを出力することとなっている。そこで、その部分に変更を行い、レスポンスパケットの情報を warning レベルで出力することとした。

具体的には、QID、ancount、nscount、arcount、（アンサーセクションの最初の RR の情報）（名前、TTL、タイプ）、A の場合は IPv4 アドレスを記録する。今回は、フィッシングを念頭に、ユーザの Web 閲覧を騙すことを想定し、A リソースレコードのみ細かく見ている。作成したパッチを以下に公開した。

```
http://member.wide.ad.jp/~fujiwara/
bind-8.4.7-IDattackLogging-00.diff
http://member.wide.ad.jp/~fujiwara/
bind-9.3.2-IDattackLogging-00.diff
```

このパッチを用いることで、BIND 8 では、security category の warning レベルに

```
UnknownID: queryID ancount/nscount/arcount
rrname IN A 192.168.1.1
UnknownID: queryID ancount/nscount/arcount
rrname IN TYPE%d
```

というログが出力され、BIND 9 では、dispatch category の warning レベルに同様のログが出力される。

このパッチによる記録の負荷は、BIND の logging 機能で querylog を記録する場合とほとんどかわらないと考えられる。Brute force attack を受けた場合はログの記録のために大きな負荷がかかることになるが、その場合は別の方法で対応すること。

3.3 実環境での調査

このパッチを、ns.tokyo.wide.ad.jp と、個人宅のフルリゾルバに仕込み、3 ヶ月間ログの観察を行った。

その結果、ns.tokyo.wide.ad.jp では約 12 万行のログを得たが、個人宅のリゾルバを含め、攻撃とわかるものは見つけられなかった。しかしながら、いくつかの特定のドメイン名で 1 つの問い合わせに対して複数の応答があることが観測された。それらは、ロードバランサの実装に DNS を用いているところのようであった。

そのうち、ある DNS サーバは、1 つの名前の問い合わせに対し、同じ QID であるが異なる IPv4 アドレスを含む 10 個程度のレスポンスを返していた。さらに、そのあとでその名前についての DNS 問い合わせを送ってきた。複数の応答のうちの 1 つがラン

ダムに選ばれることでロードバランスできることと、その結果の確認を行っているように見えた。

調査の結論として、小規模なサイトのフルリゾルバでは攻撃の検証が難しいこと、DNS を使ったロードバランサの興味深い挙動を観察することができた。

第 4 章　定期的な UDP ポートの変更による BIND9 サーバへのキャッシュ汚染防止

4.1 DNS のキャッシュ汚染攻撃

DNS のキャッシュサーバに対する攻撃として、キャッシュ汚染（cache poisoning）と呼ばれる手法がある [10]。この攻撃は一般に、キャッシュサーバが送信する問い合わせに含まれるパラメータと一致する偽の応答を正当なサーバより先に返すことによって実現される。問い合わせのパラメータの中で、攻撃者が（一般には）制御できず、したがって推測、盗聴、総当たり探索等によって一致させる必要があるものは、問い合わせ ID、送信元のポート番号、問い合わせ先のサーバのアドレスである。

与えられた問い合わせ名に対し、問い合わせ先サーバのアドレスは通常数個から高々十数個であるため、総当たり探索にかかるコストは小さい（なお、アドレスについてはさらにソースアドレスを詐称したパケットがフィルタされることなく攻撃先に到達できるという条件も必要になるが、この点は本報告書での議論の本質から外れるので考えないことにする）。一方、ID およびポート番号はともに 16 ビット整数値であり、この双方を総当たり探索によって一致させることは現実的ではないため、問い合わせパケットを盗聴してパラメータを取得できない限りキャッシュ汚染攻撃を成立させることは理論上は困難だといえる。

ところで、広く普及している ISC の BIND キャッシュサーバは、起動している期間中は固定の問い合わせ元ポート番号を用いるため、キャッシュ汚染攻撃を成功させるために一致させる必要があるパラメータは実質的に 16 ビット整数値である問い合わせ ID のみとなる（アドレスの個数についてはこれより十分に小さいので議論を簡略化するためにここでは無視する）。この仕様がキャッシュ汚染攻撃に対する潜在的な脆弱性となっていることは以前から指摘されていた。

ただし、この「脆弱性」を突いて実際に攻撃を成功させることは簡単ではない。まず、16 ビットの空間でも単純な推測に基づく方法では一致させることは困難である。また、総当たり探索による攻撃は、それ自体は成立しても、そのために生じる大量の応答パケットによって攻撃が検出されやすくなるという欠点がある。

ところが、TTL が十分に小さい問い合わせ名に対しては、より検出されにくい形でこの攻撃を成立させられる場合がある [318]。TTL が小さく、また頻繁にクライアント（スタブリゾルバ）からの問い合わせがあるような著名な名前であれば、キャッシュサーバからの問い合わせも TTL に応じて頻繁に送信される。したがって、攻撃者側は、検出されない程度の頻度で問い合わせ ID を総当たり探索し、比較的短時間のうちに高い確率で攻撃を成立させることができる。[318] では、TTL 30 秒、問い合わせ先サーバ数 2、キャッシュサーバとサーバ間の通信にかかる時間が 20 ミリ秒、攻撃側の送出レートが毎秒 10 パケットという条件で、100 台のキャッシュサーバに並行して攻撃をかけた場合、38 時間後には 50%の確率でどれか一台のサーバへの攻撃が成功するという計算結果を示している。

この問題を軽減するためのもっとも単純な対策は、問い合わせのポート番号をより推測されにくく、あるいは総当たり攻撃への耐性が高くなるように変更することである。実際、BIND 以外の DNS サーバの一部、たとえば djbdns や power DNS recurser は問い合わせごとに異なるポート番号を利用している。筆者は、BIND の最新版である BIND9 のキャッシュサーバ上で同様の機能を実現するパッチを作成した。本報告書ではこの機能の概要を紹介する。

4.2 BIND9 キャッシュサーバへのパッチ

このパッチを適用すると、問い合わせ用に複数個の UDP ソケットを開き、それぞれを異なるポートに bind する。また、各ポートは一定時間ごとに更新される。問い合わせ時には、複数のソケットのうちの 1 つをランダムに選んで利用する。問い合わせごとにポートを変更する方法を採らなかったのは、その際のソケットの廃棄と生成にかかるオーバヘッドを回避するためである。このため、短時間内の総当たり攻撃に対する効果は限定的（探索対象がソケット数の分増えるのみ）であるが、前述の通り、この

場合には大量の応答パケットによって攻撃を検出できると考えられるので、実用上の効果は同等であると判断した。

本パッチは以下の URL にて公開されている:

```
http://www.kame.net/~jinmei/
bind-9.4.0b2-portpool.patch
```

なお、パッチを作成した時点で対象としていた BIND9 のバージョンは 9.4.0b2 であるが、執筆時点での最新版である 9.4.0rc1 にもそのまま適用可能である。

また、本パッチは BIND の開発元である ISC にも送付し、リリース版に採用してもらうように働き掛けている。

4.2.1 利用方法

本パッチを利用するには、新規オプション “use-queryport-pool” を yes に指定する：

```
options {
        ...
        use-queryport-pool yes;
};
```

（既定値はは no、per view でも指定可能）

この機能が有効な場合、各 view について queryport-pool-ports（既定値は 8）個の UDP ソケットを同時に開き、それぞれをランダムな（non reserved）ポートに bind する。また、IPv4/v6 デュアルスタックの場合には、それぞれについて queryport-pool-ports 個のソケットが生成される。

また、queryport-pool-updateinterval（既定値は 15）分ごとにこれらのソケットを 1 つづつ更新する。すなわち、新しいソケットを開き、ランダムなポートに bind し直す。古いソケットは close() する。したがって、既定値を利用した場合、どのソケットも最大 2 時間で入れ替わることになる。

queryport-pool-ports および queryport-pool-updateinterval の値についても設定ファイル内で指定可能である。

4.2.2 実装上の論点

以下は、本機能の実装に関する主要な論点である。これらについては、今後の議論・試験運用の結果等を通じて方針を定める予定である。

1. queryport-pool-ports と queryport-pool-updateinterval の妥当な既定値は何か。現状の実装で利用している値にはとくに根拠はなく、実際の攻撃シナリオや問い合わせパターンによってはよりよい値が他にある可能性もある。たとえば、本パッチが想定しているような、比較的低レートの攻撃を防止するためには、queryport-pool-ports は 1 でも十分ということも考えられる。
2. IPv4 と IPv6 では挙動を変える（こともできるようにする）べきか。現在の実装では、queryport-pool-ports および関連するパラメータは IPv4/IPv6 共通に作用する。
3. 異なる view で用いるソケットを共有可能にすべきか。現在の実装では、use-queryport-pool を有効にしている BIND9 view においては、どの view のどのソケットも異なるポートを利用するようになっているため、多数の view を利用する設定ではその分多くのソケットが必要となる。一方、BIND9 の実装では view ごとに異なるキャッシュを持つため、複数の view で同一のソケット（ポート）を共有していてもキャッシュ汚染防止という目的には支障はなく、システム資源の有効利用という観点からは共有する方が望ましい。ただし、そのためには実装が多少複雑化すると予想される。

第 5 章　多地点での DNS トラフィックの収集および解析

本報告書は、WIDE DNS ワーキンググループならびに関連するワーキンググループなどにおいて現在進行中である DNS データ計測に関するプロジェクトに関してまとめる。

5.1 NeTraMet

WIDE mawi ワーキンググループ、UCSD/CAIDA と協調して、慶應大学ならびに東京大学から Root DNS サーバ群に対する DNS トラフィックの計測を行っている。この結果は `http://dnstap.nc.u-tokyo.ac.jp/NeTraMet/`にて公開している。

5.2 dsc

UCSD/CAIDA によって開発された dsc というソ

フトウェアを用いて、DNS トラフィックの分析を行っている。まだ試験的に行っている段階であり、ns-wide.wide.ad.jp に対する DNS トラフィックを、NTT 大手町の Verio への接続点にて計測している。

5.3 Root DNS データ収集

OARC と UCSD/CAIDA が主導をとって行われる “Day in the Life” Internet gathering イベントに参加して、Root DNS サーバ各拠点におけるトラフィック収集を行った。データの共有や分析などはこれからの課題となっている。

5.4 リゾルバ DNS サーバの計測

DNS ワーキンググループ内部の議論にて上がってきた話題として、DNS サーバへのキャッシュ汚染攻撃がある。よく問い合わせがある名前で、かつ TTL が短い名前を狙った誕生日攻撃は成功しやすいという議論がなされた。また、疑わしいと思われる実例があるとの報告もなされた。そのため、この汚染を目的とした攻撃がどの程度の間隔で行われているかを計測するために、ユーザからの大量の問い合わせがあり、かつある程度共通の問い合わせがあるような DNS サーバにてトラフィック計測を行うことを計画している。現在、DNS サーバの候補選定中であり、実現に向けて活動を進めている。

第 6 章　BOF での議論のまとめ

6.1 はじめに

本報告書は、DNS ワーキンググループが開催した BoF において、議論ならびに報告の行われた事項に関して、まとめをおこなったものである。本報告書は、以下にあげる情報に関してのまとめである。

- DNS Amplifier
- ID prefixion type brute force DNS cache poisoning (JPRS fujiwara)
- DNS のアタック事例

6.2 DNS Amplifier

JPRS の民田さんより、DNS を悪用した DoS 攻撃について発表があった。やり方としては、特定の権威サーバに大きいサイズのデータを置き、世界中のリゾルバサーバに対して問い合わせを送る。このレコードが当該リゾルバサーバにキャッシュされた後、ソースアドレスを偽造してこれらのサーバに問い合わせを送ると、偽造されたアドレスのもとに大量の返答が送り返されて DoS 攻撃が可能となる。実測では 2 G から 20 Gbps のトラフィックを発生させることができた。

最近ではリゾルバサーバがあらゆるホストからの問い合わせを受ける設定になっていないため攻撃用のリゾルバサーバを確保するのは難しいが、権威サーバを使っても似たようなことは可能である。問い合わせに対して極端に大きいサイズの返答を返してくる権威サーバは DoS 攻撃用の増幅器として使うことができる。

これに対して、リゾルバサーバはアクセス制御をしっかり行うとして、権威サーバが使われるのは防げないのでは、Rate Control などを利用するぐらいしかない、gray list を使えばいいのでは、などの意見が出された。

6.3 ID prefixion type brute force DNS cache poisoning (JPRS fujiwara)

JPRS の藤原さんより、DNS キャッシュ攻撃についての発表があった。

DNS は、アプリケーション (1)、リゾルバライブラリ (2)、リゾルバサーバ (3)、権威サーバ (4) の順番で問い合わせが送信され、返答はこの逆の順番で返信されてくる。リゾルバライブラリおよびリゾルバサーバは、問い合わせと返答のクエリ ID、QNAME、QCLASS、QTYPE といったフィールドをチェックし、送信した問い合わせに対応する返答を識別する。そのため、返答のパケットを偽造し、正規の応答より先に返答することによって、DNS の返答を詐称できる。

攻撃にはいくつかの方法があるが、今回は Brute Force Attack による cache poisoing についての発表を行った。

Brute Force Attack は、ターゲットとなるリゾルバサーバに対してクエリ ID、ポートなどを変えつつ返答を大量に送る。そのうちのひとつが問い合わせに対応する返答になれば、返答を詐称でき、偽造した返答の cache をリゾルバサーバに持たせることができる。

発表中に、1 台のリゾルバサーバに対して 10 個の偽情報の注入を試みる攻撃のデモを行った。攻撃の結果、注入を試みた 10 個の偽情報のうち半分程度の注入が成功したことが確認できた。

この攻撃に対する検出方法として、DNS トラフィックの急激な増加を検知する、ネームサーバの統計情報から問い合わせと返答の数の違いを調べる、送ってない query に対する返答などを検知するといった方法が紹介された。

これに対して、ID 空間が 16 bit なのが問題ではないか、UDP のポートを変えないのも問題、ポートを変えないのはパフォーマンスの関係から、DNS パケットに cookie などを入れるとより耐攻撃性が上がるのでは、などの意見が出た。

6.4 DNS のアタック事例

JPRS の民田さんより、DNS のアタック事例について発表があった。とある有名ブログサイトが他のサイトに誘導されており、調査したところ ISP のリゾルバサーバに不正なレコードのキャッシュが入っていた。これは正しいデータとは異なり、またキャッシュの TTL も 5 日と大きく、誘導先は何かの宣伝サイトだったため、攻撃の疑いが高かった。

今回のようなアクセスを引き寄せるための攻撃の場合、特定のリゾルバサーバを狙う必要が無いため無差別にやればよく、かつそれぞれに対して集中して DNS のパケットを送らなくてよいため検知がしにくいという問題がある。また、多くの著名サイトでは負荷分散や障害時対応のため、DNS レコードのキャッシュを短めにしているということもあり、それも攻撃が成功しやすい原因となっている。

これらに対して、やはり cookie が有効なのではないか、あまりに短い TTL はリゾルバサーバ側で無効にする必要があるのではないか、サイトを運用している側からすれば DNS のキャッシュはできるだけ短くしたい、プロトコル・実装・オペレーションのそれぞれの側面で攻撃成功率をより低くする努力をする必要があるなどの意見が出た。

第 7 章　まとめ

今年度は DNS に対する攻撃に関連した議論と対策についての開発を行った。今後はこれらの研究成果による運用に関する知見、および実装の普及を目指すものとする。また、プロトコル自体の改良も視野に入れた根本的な対策の研究も今後の重要な課題のひとつである。