

JPRS トピックス&コラム



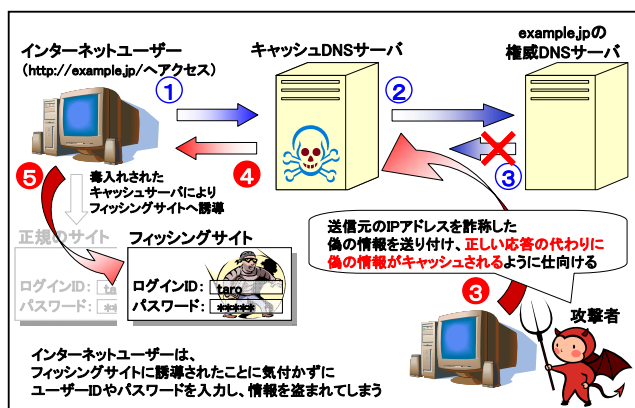
新たなるDNSキャッシュポイズニングの脅威 ～カミンスキー・アタックの出現～

DNSに不正なデータを記憶させドメイン名の乗っ取りを図る「DNSキャッシュポイズニング」と呼ばれる攻撃の危険性が高まっています。この攻撃の概要と対策について解説します。

■キャッシュ機能とDNS キャッシュポイズニング

DNS には、問い合わせによって得られた IP アドレスや権威 DNS サーバ名などの情報を一時的に記憶し、名前解決の際にそれらを再利用することで処理の効率化を図る、という機能があります。これを DNS の**キャッシュ機能**といい、キャッシュ DNS サーバや一部の DNS クライアントに実装されています。キャッシュ機能は DNS サーバへの問い合わせ数を大幅に減少させ、DNS やネットワークの負荷を全体に軽減させるという重要な役割を担っています。

しかし、この機能を悪用し、偽のデータをキャッシュに記憶させることでドメイン名の乗っ取りやフィッシングなどを図る「**DNS キャッシュポイズニング** (DNS cache poisoning)」と呼ばれる攻撃が知られています。



DNS キャッシュポイズニングの概要

「ポイズニング」は、毒性を持つものが内部に取り込まれることにより正常な機能が阻害されることを表しており、「毒入れ」とも呼ばれています。

■DNS キャッシュポイズニングの仕組み

DNS では主要な通信プロトコルとして UDP を使用します。UDP では TCP と異なり、相手との通信前の折衝やデータの到達確認を行いません。そのため、UDP は

TCP に比べてデータ到達の信頼性は下がりますが、相手との通信に必要な手間が少ないため、一度のやりとりで通信が完了し、かつサーバにおいて数多くのクライアントからの要求をできる限り短時間で処理する必要のあるプロトコルに向けたものであるといえます¹。

しかし、UDP はその特性から、TCP に比べ通信データの偽造が容易であるという短所があります。そのため DNS では、処理ごとに毎回異なった「ID」を付け、問い合わせ内容とともに ID の一致も併せてチェックすることにより、応答の正当性を確認しています。

しかし、現在の DNS プロトコルでは、悪意を持った第三者が、

- ①発信元として DNS サーバの IP アドレスを偽装し、
 - ②問い合わせに使われた UDP ポートと同じポートに、
 - ③問い合わせに使われた ID と同じ ID を付け、
 - ④本来の DNS 応答よりも先に応答を返す
- ことができた場合、問い合わせ元はそれが偽のデータであることを判別できないため、DNS キャッシュポイズニングが成立してしまいます。

■DNS キャッシュポイズニングと TTL 値の関係

ここまでで解説した DNS キャッシュポイズニングの脆弱性そのものは、以前から知られていました。従来の DNS キャッシュポイズニング攻撃では、通常の DNS 応答、例えば www.example.jp に対し攻撃を試みる場合、www.example.jp の問い合わせに対する応答そのものを偽造する形で行われていました。

この場合、DNS キャッシュポイズニングが成功する可能性は、該当するデータの有効時間 (TTL: Time To

¹ DNS が 20 年以上もの間その基本仕様を大きく変更することなく、急成長したインターネットを現在まで支え続けてこれたのは、通信プロトコルとしてコストの少ない UDP が採用されていたことが大きな要素の一つであった、といえるでしょう。

Live)の設定値に依存します。つまり、キャッシュが有効である間は目的とする名前に対する外部への問い合わせが行われないため、権威 DNS サーバ側で長い TTL を設定することにより、DNS キャッシュポイズニングに関する危険性をある程度軽減することができます。

TTLが有効な間、キャッシュDNSサーバは
その名前の外部ネットワークへの検索を行わない

TTLが有効な間はそのキャッシュDNSサーバに対し、
従来の方法でのDNSキャッシュポイズニングはできない

DNS キャッシュポイズニングと TTL の関係

■新たに発見された攻撃方法

しかし、セキュリティ研究者のダン・カミンスキー氏が、DNS キャッシュポイズニングをより効率的に実行可能な新しい攻撃方法を発見し、その方法が2008年の7月に明らかにされたことから、DNS キャッシュポイズニング攻撃に対する危険性が**従来に比べ急激に高まりました**。この方法は「**カミンスキー・アタック** (Kaminsky Attack、あるいは Kaminsky's attack)」と呼ばれています。

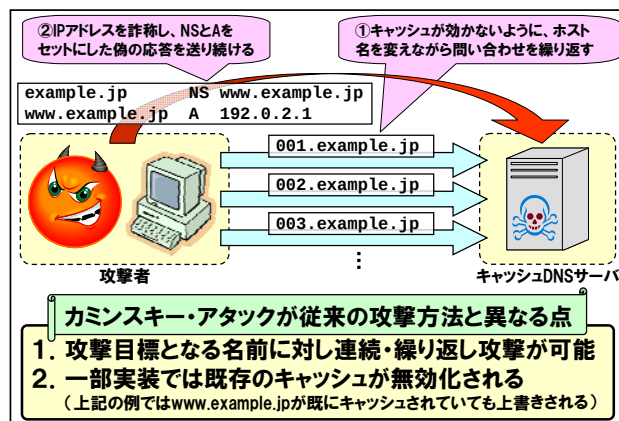
カミンスキー・アタックでは、攻撃対象の名前(例: www.example.jp)と同じドメイン名内の存在しない名前(例: 001.example.jp)の問い合わせを攻撃目標となるキャッシュ DNS サーバに対して送り(あるいは、その名前を検索させるように仕向け)、その直後に「私はその名前(001.example.jp)を知らないが、www.example.jp が知っている。その IP アドレスは xxx.xxx.xxx.xxx (攻撃者が用意した偽のサーバの IP アドレス)である」という偽の応答情報を、ID を変化させながらキャッシュ DNS サーバに大量に送り付けます²。もし、この偽の応答が前節で解説した条件を満たしている場合、DNS キャッシュポイズニングが成立してしまいます。

現在の DNS プロトコル(RFC 1035)では、DNS の ID は16ビットと定められているため、IDは最大でも65,536通りとなります。この値は現状のインターネットにおいて総当たり攻撃に対し必ずしも十分な耐性を有しているとはいえません。

² 攻撃者は偽のサーバ上で、example.jp の偽の権威 DNS サーバをあらかじめ動作させておきます。

また、カミンスキー・アタックでは、攻撃者が問い合わせる名前を毎回変化させることにより、攻撃目標となる名前に対する攻撃を**連続して繰り返し行うことができる**ようになります。従来のDNSキャッシュポイズニングではポイズニングが一度失敗した直後に同じ名前に対して即座に再攻撃を試みることは不可能でしたが、カミンスキー・アタックではそれが可能となります。

更に、一部のキャッシュ DNS サーバの実装ではカミンスキー・アタックで攻撃を受けた場合、通常の名前検索で攻撃対象の名前が既にキャッシュされている場合であっても、それを無視する形で偽の情報が上書きされてしまうことから、攻撃対象データの TTL の設定値やキャッシュ情報の有無にかかわらず、外部からの DNS キャッシュポイズニングが成立してしまうことがあります。



カミンスキー・アタックの概要

■DNS キャッシュポイズニングの脅威

DNS キャッシュポイズニングにより偽のデータを記憶させられてしまった場合、その被害はそのキャッシュ DNS サーバを参照しているすべてのクライアントに及びます。また、DNS はほぼすべてのインターネットサービスが使用しているため、DNS キャッシュポイズニングはインターネット全体の安全性にかかわる問題です。

特に、今回のカミンスキー・アタックは危険性が非常に高く、インターネット全体にとって大きな脅威であり、早急な対策が必要になります。

次節では、カミンスキー・アタックを含む、DNS キャッシュポイズニングへの対策方法について解説します。

■DNS キャッシュポイズニングへの対策方法

1. 問い合わせ UDP ポートのランダム化…必要に応じサーバソフトウェアのバージョンアップを

従来の主なキャッシュ DNS サーバの実装では、問い合わせに使う UDP ポートは一つに固定されているか、あるいは決められた範囲の限られたポートしか使用しないものがほとんどでした。これを問い合わせごとに毎回変える(ランダム化)ように変更することで、総当たり攻撃に対する耐性を高めることができます。そのため今回、問い合わせ UDP ポートをランダム化するための緊急パッチが、各開発元からリリースされました。

UDP ポートをランダム化することにより、DNS キャッシュポイズニングが成功する確率を、ID のみがランダムでポートが固定である場合に比べ、大幅に低下させることができます。

あるキャッシュDNSサーバに対し、DNSキャッシュポイズニングが成立する確率

$$P_s = \frac{R \times W}{N \times Port \times ID}$$

R: 攻撃対象1台あたりに送るパケット量(pps)

W: 攻撃可能な時間(問い合わせ⇒応答のRTT)

N: 攻撃対象レコードを保持する権威DNSサーバの数

Port: 問い合わせに使われるUDPポート番号の数

ID: DNSのID (16bit = 65,536)

上記式において、

➢UDPポート番号が固定 ⇔ Port = 1

➢UDPポート番号がランダム ⇔ Port = 65,536

となるため、UDPポート番号をランダム化することにより、DNSキャッシュポイズニングが成立する確率 P_s を小さくすることができる。

UDP ポート番号ランダム化の効果

カミンスキー・アタックでは問い合わせ UDP ポートが固定であった場合、**数秒程度の攻撃でDNS キャッシュポイズニングを成立させることができます**。キャッシュ DNS サーバを運用されている方は現在使用しているサーバソフトウェアのバージョンを確認し、必要に応じ最新版に更新しておきましょう。

主な開発元における対応情報は、後述の CERT/CC の Web ページにまとめられています。

US-CERT Vulnerability Note VU#800113
<<http://www.kb.cert.org/vuls/id/800113>>

2. オープンリゾルバは危険…適切なアクセスコントロールやフィルタリングの適用を

インターネット上のどこからのDNS再帰検索要求でも受け付ける状態になっているキャッシュ DNS サーバを「オープンリゾルバ」といいます。オープンリゾルバはDNS Amp 攻撃³の踏み台に使われる危険性があるなど、本来好ましくない設定ですが、DNS キャッシュポイズニングの攻撃者から見た場合オープンリゾルバには、**攻撃者が任意のタイミングで、任意のドメイン名に対するキャッシュポイズニング攻撃を始めることができるため、非常に危険な状態です**。

また、DNS サーバでアクセスコントロールが適切に設定されていても、発信元 IP アドレスを偽装した DNS データが外部から到達可能であった場合、そのデータを受け取ったキャッシュ DNS サーバが反復検索を始めてしまう恐れがあります。

そのため、DNS キャッシュポイズニングを防ぐためにはDNSサーバにおける対策だけでは不十分であり、内部ネットワークの IP アドレスを詐称したデータが外部から到達することがないように、ネットワークにおいてもフィルタリングなどの適切な対策を取る必要があります。

3. 権威 DNS サーバは狙われやすい…機能分割と反復検索機能の無効化を

権威 DNS サーバは、それぞれのドメイン名を管理するためのサーバとしてインターネット上に広く公開されます。そして、公開が目的である権威 DNS サーバの IP アドレスは誰でも簡単に知ることができるため、もし権威 DNS サーバがオープンリゾルバの状態に設定されていた場合、非常に危険です。

本来、権威 DNS サーバとキャッシュ DNS サーバはDNSの構成上別の機能です。そのため、権威 DNS サーバとキャッシュ DNS サーバはできる限り別のサーバ(あるいは別の IP アドレス)に分割し、権威 DNS サーバでは反復検索機能を無効に設定しておくことが推奨されています。

³ DNS Amp 攻撃の概要と対策については、JPRS トピックス&コラム No.003「DDoS にあなたの DNS が使われる」をご参照ください。

■根本的な解決方法は DNSSEC の導入

これまでに解説した方法は、DNS キャッシュポイズニングの危険性を確率的に下げたためのものです。

DNS キャッシュポイズニングを根本的に防ぎ、インターネットの安全性を向上させるためには、DNS プロトコル自体の改良が必要になります。そして、それを目的として開発された DNS のセキュリティ拡張機能である **DNSSEC⁴** の導入が進められ始めています。

しかし、DNSSEC を導入するためには、関連する DNS サーバ(権威 DNS サーバ、キャッシュ DNS サーバの双方)すべてを DNSSEC 対応のものに更新する必要があるなど、インターネットを構成するさまざまな立場から対応を進めていく必要があります。このため、DNSSEC の普及促進のための活動を今後どう進めていくかが、大きな課題となっています。

■攻撃の検知と防御—現時点における試み

このように根本対策である DNSSEC の普及には今後ある程度の時間を要するため、DNS キャッシュポイズニング攻撃を効率良く検知し、効果的な防御を行うための手法が研究開発・発表され始めています。

カミンスキー・アタックを始めとする DNS キャッシュポイズニング攻撃では、通常の運用ではほとんど検出されない、問い合わせ時の ID や UDP ポート番号と一致しない、不正な DNS 応答が観測されます。これを効率良く検知することで攻撃をブロックしたり、必要に応じより信頼性の高い TCP での再問い合わせを行ったりするなど、さまざまな対策が提案、実装されています。

根本対策である DNSSEC も、JP ドメイン名を含むいくつかの TLD や RIR など導入が始まっています⁵。

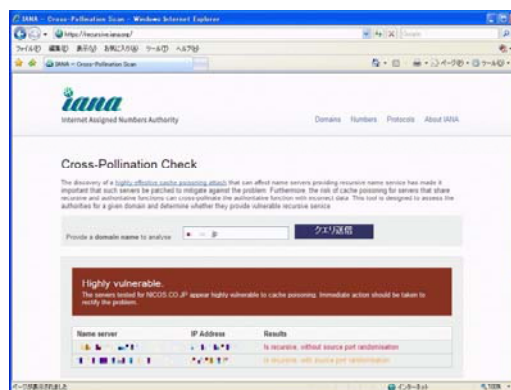
■自分の DNS サーバをチェックするには

ICANN/IANA では、DNS の管理者が権威 DNS サーバの設定状況をチェックするための Web ページを開設しています。

この Web ページでは、①指定するドメイン名の権威

DNS サーバがオープンリゾルバになっていないか、②もしオープンリゾルバになっている場合、問い合わせ UDP ポートが固定されていないか、の2点をチェックすることができます。

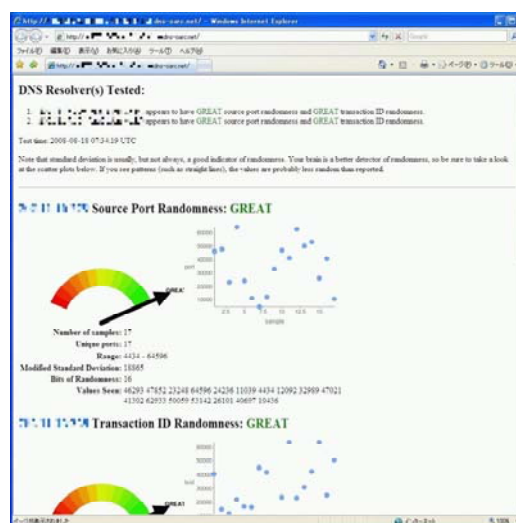
IANA — Cross-Pollination Scan
<<https://recursive.iana.org/>>



チェック結果の出力例(IANA)

また、DNS に関する調査・分析活動や情報交換の場を提供している DNS-OARC (DNS Operations, Analysis, and Research Center) では、ユーザーが現在使っているキャッシュ DNS サーバの ID と UDP ポート番号が十分なランダム性を備えているかどうかを視覚的にチェックできる Web ページを開設しています。

Web-based DNS Randomness Test | DNS-OARC
<<https://www.dns-oarc.net/oarc/services/dnsentropy/>>



チェック結果の出力例(OARC)

⁴ DNSSEC の概要と普及に向けた課題については JPRS トピックス&コラム No.013「DNSSEC の概要と今後の展開」をご参照ください。

⁵ JPRS では 2011 年 1 月 16 日に DNSSEC を JP ドメイン名に導入しました。