ちょうどいいDNSの設定と運用のために必要なことを考える

～ランチのおともにDNS～

2024年11月26日

Internet Week 2024 ランチタイムセミナー

株式会社日本レジストリサービス（JPRS）

森下 泰宏・熊谷 維魅

講師自己紹介

- 森下 泰宏（もりした やすひろ）
  - 所属：JPRS 技術広報担当・技術研修センター
  - 主な業務内容：技術広報活動全般・社内外の人材育成
  - 一言：**久々に浅草橋ヒューリックホールの壇上に立っています！**

- 熊谷 維魅（くまがい いみ）
  - 所属：JPRS システム部
  - 主な業務内容：ネットワークの構築運用・DNS関連イベントの講師など
  - 一言：**発表や講演で話すことが好きなので、今日も全力で頑張ります！**

本日の内容

1. DNSにおける上限値の状況（話者：熊谷）

2. これまでに報告された脆弱性の事例（話者：森下）

3. ちょうどいいDNSのために必要なこと（話者：森下・熊谷）

「**上限値**」をキーワードとして、ちょうどいいDNSを
実現するために必要なことについて考えていきます

1. DNSにおける上限値の状況

DNSプロトコルにおける上限値

- DNSには、**プロトコルにおける上限値**がいくつか設定されている
  - DNSが開発された1980年代の状況から、プロトコルにおいて具体的な上限値が設定されている項目は多くない

項目	上限値
メッセージサイズ	UDP：512バイト → EDNS0によるネゴシエーション TCP：65,535バイト
ラベルの長さ	63文字
ドメイン名の長さ	253文字（最後のドットを含まない）
リソースレコードタイプ	65,536種類
問い合わせコード	16種類
応答コード	16種類 → EDNS0により65,536種類に拡張
メッセージ内の各セクションの数	65,535個

DNSプロトコルにおける上限値の例

上限値が設定されていない項目

- そのため、DNSには**具体的な上限値が設定されていない項目**が存在する
  - 以降では、**実装・運用で制限される、以下の項目の上限値の状況**を解説する

項目	上限値
一つのゾーンのレコードの数	プロトコル上の上限値は存在しない （例：.comゾーンには1億以上の委任が存在）
一つのゾーンから委任されるゾーンの数	
サブドメインの深さ	ラベル・ドメイン名の長さで制限される
名前解決におけるCNAMEレコードの段数	**実装・運用で制限**される
名前解決におけるUnrelatedな委任の段数	
RRSetごとのレコードの数	
ゾーンごとのネームサーバーホスト名の数	メッセージサイズで制限される **実装・運用で制限**される場合もある
ゾーンごとのDNSSEC鍵の数	
レコードごとのDNSSEC署名の数	

これらの項目を解説

具体的な上限値が設定されていない項目の例

名前解決におけるCNAMEレコードの段数

- リゾルバーやサービスの**実装で段数を制限**している
  - 主なリゾルバー実装：10～20段（BIND 9：16段、Unbound：11段）
  - AWS証明書マネージャーのDNS検証：4段（5段以上は検証エラー）

> AWS Certificate Manager DNS validation - AWS Certificate Manager
> <https://docs.aws.amazon.com/acm/latest/userguide/dns-validation.html>

- RFC 1034には、以下のように記述されている

> domain software **should not fail when presented with CNAME chains or loops**;
> CNAME chains should be followed and **CNAME loops signaled as an error**.

> ドメインソフトウェアは**CNAMEの連鎖やループに遭遇しても障害に陥るべきではない。**
> CNAMEの連鎖は追随されるべきであり、**CNAMEのループはエラーとして通知されるべきである。**

名前解決におけるUnrelatedな委任の段数

- Unrelatedなネームサーバーホスト名のみで構成される委任情報を名前解決する場合、**要求された名前解決をいったん保留し、ネームサーバーホスト名を名前解決する**必要がある

※Unrelatedの意味については、2023年のランチセミナーの資料を参照
（例：example.jpゾーンのネームサーバーホスト名がns1.example.com）

- Unrelatedな委任の段数が多いと**名前解決のコストが上昇する**ため、多くのフルリゾルバーの**実装で段数を制限**している

```
$ORIGIN jp.
@               86400 IN SOA ...
...
example.jp.     86400 IN NS ns1.example.com.
                86400 IN NS ns1.example.net.
```

Unrelatedなネームサーバーホスト名のみで構成される委任情報の例

RRSetごとのレコードの数（1/2）

- 上限値は設定されておらず、**メッセージサイズにより制限**される
  - 「**最大積載量：積めるだけ**」の状態
  - **登録システムやDNSサービスなどの実装によっても制限**される
- 多数のレコードを含むRRSetの応答を、**サイバー攻撃の元ネタに利用される事例**が観測されている
  - DNSリフレクター攻撃の元ネタ
  - 権威DNSサーバー・フルリゾルバーを攻撃する元ネタ

```
$ORIGIN example.jp.
@       86400 IN SOA ...
www     300   IN A 192.0.2.1
        300   IN A 192.0.2.2
        300   IN A 192.0.2.3
```

3個のAレコードで構成される
www.example.jpのA RRSetの例

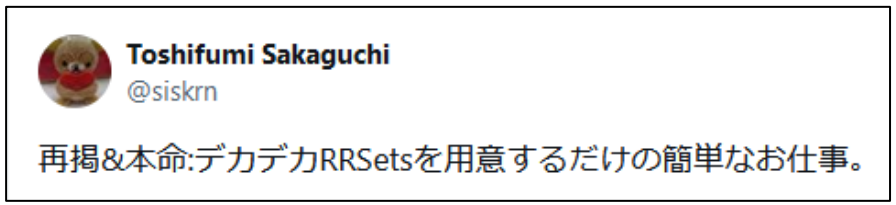
RRSetごとのレコードの数（2/2）

- DNSリフレクター攻撃の元ネタの例
  - **238個のAレコードで構成され、3,889バイトの応答を返すRRSet**が DNSリフレクター攻撃の元ネタに使われた事例が観測された

IW2013：DNSのメッセージサイズについて考える～ランチのおともにDNS～（29ページ）
https://jprs.jp/tech/material/iw2013-lunch-L3-01.pdf#page=29

- 権威DNSサーバー・フルリゾルバーを攻撃する元ネタの例
  - 2024年7月に報告された**CVE-2024-1737**で、BIND 9ではRRSetごとのレコードの数の**上限値が100に制限**された

CVE-2024-1737: BIND’s database will be slow if a very large number of RRs exist at the same name
<https://kb.isc.org/docs/cve-2024-1737>



ゾーンごとのネームサーバーホスト名の数

- 上限値は設定されておらず、**メッセージサイズにより制限**される
  - 「**最大積載量：積めるだけ**」の状態
  - **登録システムやDNSサービスなどの実装によっても制限**される
- ルートサーバーはプライミング[*1]に配慮し、**13系列に設定**されている
  - 14系列でも運用できたが、将来の拡張の余地を残すために13系列にされた

> According to @agercasa (Jaap Akkerhuis), **Bill Manning said they wanted to be conservative and leave some room for future expansion.**

> Why 13 DNS root servers? :: Miek Gieben
> <https://miek.nl/2013/november/10/why-13-dns-root-servers/>

[*1] フルリゾルバーがルートゾーンのネームサーバー情報をルートサーバーに問い合わせて、自身の情報を更新すること。

ゾーンごとのDNSSEC鍵の数・レコードごとのDNSSEC署名の数

- 上限値は設定されておらず、**メッセージサイズにより制限**される
  - 「**最大積載量：積めるだけ**」の状態
- DNSSECは**一つのゾーンに複数の鍵を設定でき**、かつ、**一つのRRSetに複数の署名を追加できるように設計**されている
  - ZSKとKSKの分離、鍵のロールオーバー、マルチ署名者DNSSEC（RFC 8901）などで利用されている
- この設計が、2024年2月に発表された**KeyTrap**に利用された

JPRS用語辞典｜KeyTrap（キートラップ）
<https://jprs.jp/glossary/index.php?ID=0276>

このパートのまとめ

- ここまでで紹介した**実装・運用における上限値**はいずれも、**DNSを安定稼働させるための方策**の一つとして設定されている
  - 名前解決で **過負荷にさせる** ⇒ 上限値を設定して **負荷を下げる**
  - 名前解決を **ループさせる** ⇒ 上限値を設定して **ループを回避する**
  - 名前解決で **サイバー攻撃させる** ⇒ 上限値を設定して **攻撃の効率を下げる**

  プロトコルの上限値が緩やかな部分を、**実装・運用でカバー**している
- 次のパートでは、上限値の設定が十分でなかったことで発生した、**いくつかの脆弱性の事例を紹介・解説**する

2. これまでに報告された脆弱性の事例

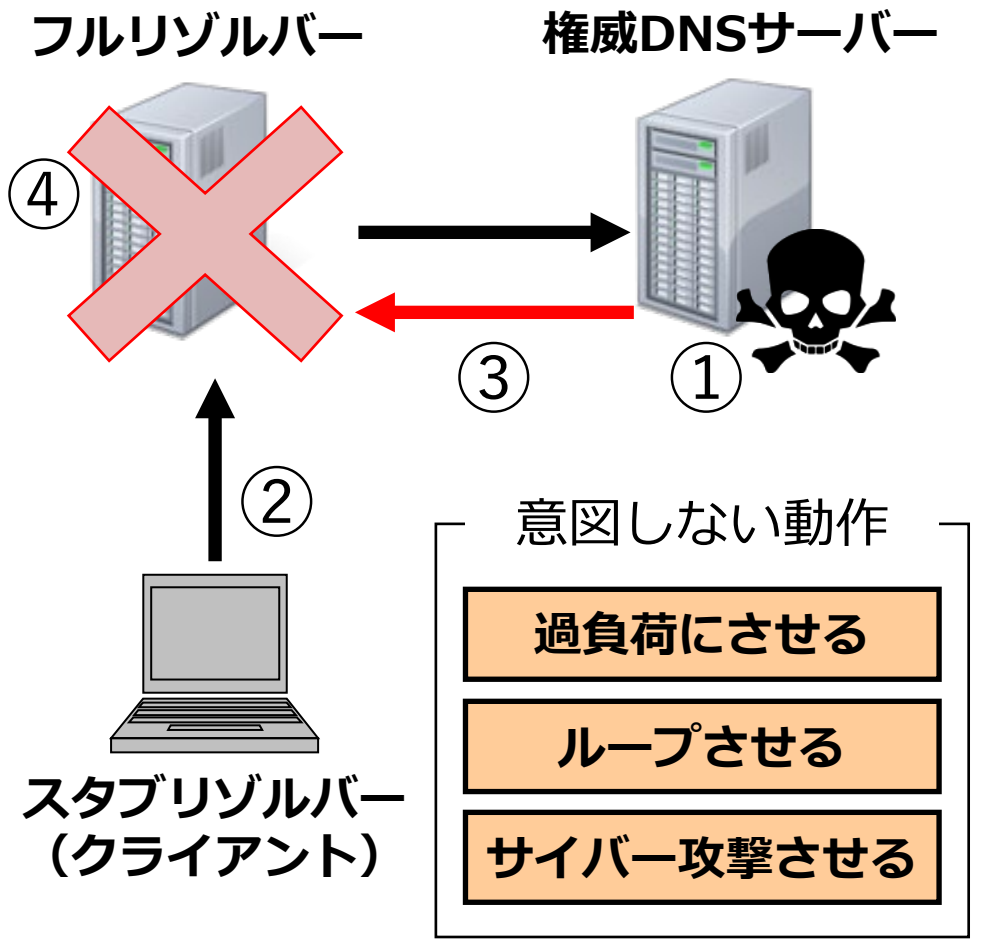
このパートで紹介・解説する脆弱性

- **iDNS Attack**（2014年）
- **NXNSAttack**（2020年）
- **TsuNAME**（2021年）
- **KeyTrap**（2024年）

いずれも、**権威DNSサーバーを使ってフルリゾルバーを攻撃する**脆弱性

脆弱性を発生させる仕組み

- いずれも、**権威DNSサーバーに攻撃用のデータを設定して名前解決させ、フルリゾルバーに注入することで意図しない動作をさせる**攻撃手法が用いられている
  - ① 攻撃用のデータを**設定**する
  - ② そのデータを**名前解決**させる
  - ③ 攻撃用のデータが**注入**される
  - ④ **意図しない動作**に陥る



iDNS Attack
(CVE-2014-8500、CVE-2014-8601、CVE-2014-8602)

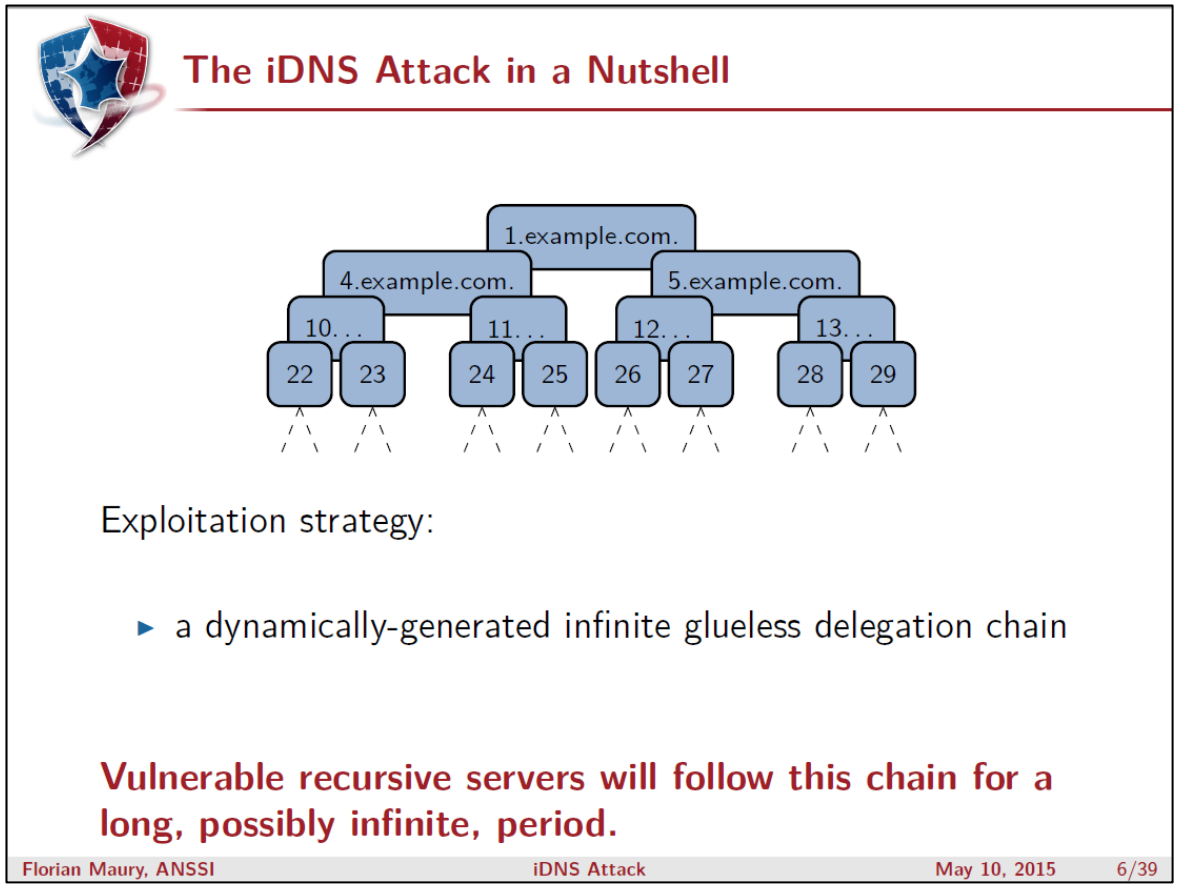
- 2014年に報告された脆弱性
  - **複数のフルリゾルバー実装**が脆弱性の対象

The Infinitely Delegating Name Servers (iDNS) Attack | ANSSI
<https://cyber.gouv.fr/publications/infinitely-delegating-name-servers-idns-attack>

- 2種類の攻撃手法が含まれている **過負荷にさせる** **サイバー攻撃させる**
  - **多段の委任情報をたどらせて、フルリゾルバーを過負荷にする**
  - グルーレコードに**標的のIPアドレスを設定して、フルリゾルバーに攻撃させる**

iDNS Attackの仕組み

- **1**.example.comのNSに**4**.example.comと**5**.example.comを設定
- **4**.example.comのNSに**10**.example.comと**11**.example.comを設定
- **5**.example.comのNSに**12**.example.comと**13**.example.comを設定
- **10**.example.comのNSに**22**.example.comと**23**.example.comを設定
- **これを延々と繰り返す（右図）**
- **1.example.com**を名前解決させ、**多段の委任情報を処理させる**

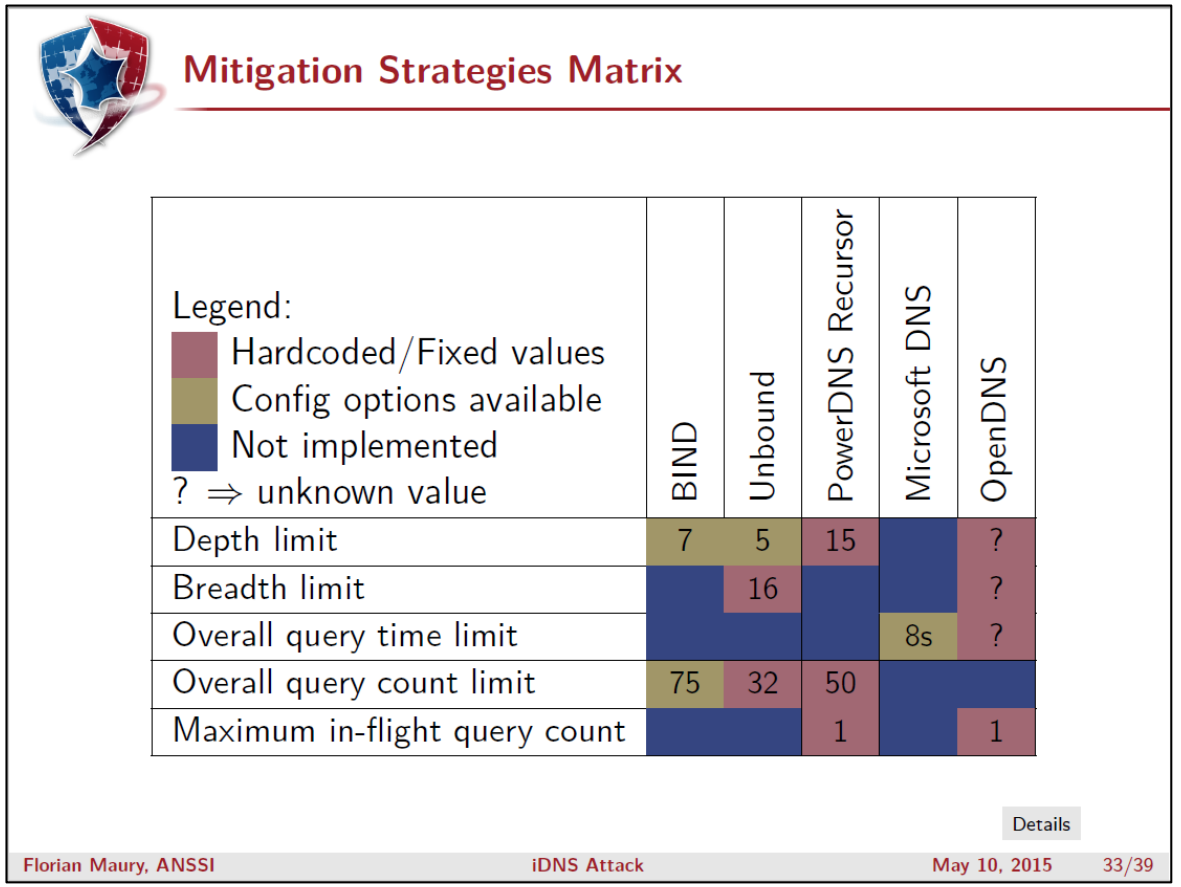


引用元：<https://indico.dns-oarc.net/event/21/contributions/301/attachments/272/492/slides.pdf>

iDNS Attackの対策

- 名前解決の動作に**上限値を導入**
  - 委任の深さ
  - 委任応答のNSの数
  - 問い合わせの時間
  - 問い合わせの回数
  - 実行中のクエリ数

負荷を下げる　**攻撃の効率を下げる**



Mitigation Strategies Matrix

Legend: Hardcoded/Fixed values; Config options available; Not implemented; ? ⇒ unknown value	BIND	Unbound	PowerDNS Recursor	Microsoft DNS	OpenDNS
Depth limit	7	5	15		?
Breadth limit		16			?
Overall query time limit				8s	?
Overall query count limit	75	32	50		
Maximum in-flight query count			1		1

Details

Florian Maury, ANSSI　iDNS Attack　May 10, 2015　33/39

引用元：<https://indico.dns-oarc.net/event/21/contributions/301/attachments/272/492/slides.pdf>

NXNSAttack
（CVE-2020-8616、CVE-2020-12662、CVE-2020-12667）

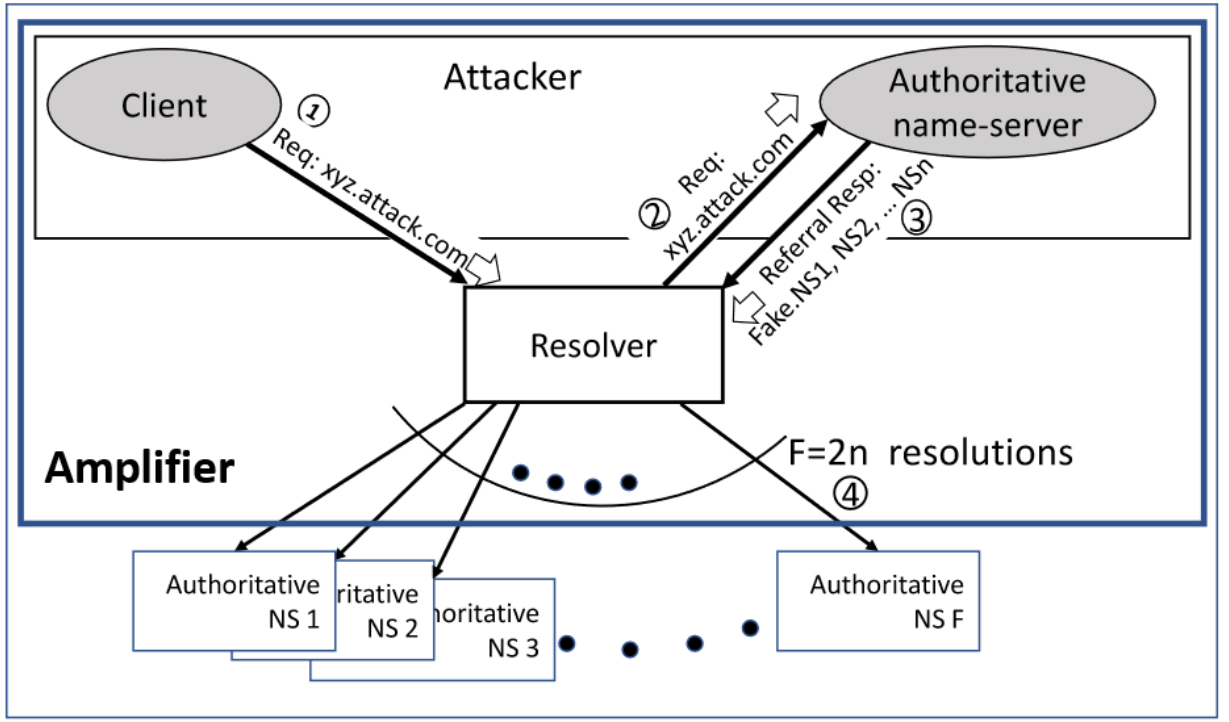
- 2020年に報告された脆弱性
  - 複数のフルリゾルバー実装が脆弱性の対象

NXNSAttack - NXNSAttack by Lior Shafir, New DNS DDoS vulnerability
<https://www.nxnsattack.com/>

- 2種類の攻撃手法が含まれている　**過負荷にさせる**　**サイバー攻撃させる**
  - **一度に多数の委任情報を処理させて、フルリゾルバーを過負荷にする**
  - 多数のNSレコードに**標的のドメイン名の存在しないサブドメインを設定して、フルリゾルバーにそのドメイン名の権威DNSサーバーを攻撃させる**

NXNSAttackの仕組み

- 攻撃用のドメイン名を準備し、**多数のNSを設定する（右図）**
  - **ドメイン名圧縮**を利用することで、多数のNSを設定可能
- 攻撃用のドメイン名を名前解決させて**多数のNSを一度に処理させ、過負荷にする**



引用元：<https://www.nxnsattack.com/>

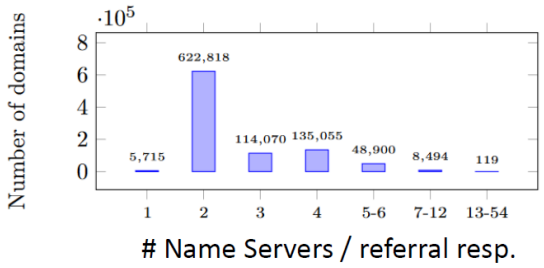
NXNSAttackの対策

- 名前解決の動作に**上限値を導入**
  - 一度に処理するNSの数
  - 委任応答のNSの数
- 名前解決の**動作を工夫**
  - 存在しないNSへの応答を検知
  - DNSSEC不在応答を活用

負荷を下げる　**攻撃の効率を下げる**

Mitigation

- MaxFetch(k) – Resolve NS-names k at a time, not all at once
  - Amortized on several queries
- MaxBreadth – bound # of NS-names per referral response
- Detect NX NS replies (NLnetLabs)
- DNSSEC – NSEC (Petr Špaček)



引用元：<https://www.usenix.org/system/files/sec20_slides_afek.pdf>

TsuNAME

- 2021年に報告された脆弱性
  - 複数のパブリックDNSサービスが脆弱性の対象

TsuNAME - Vulnerability that can be used to DDoS DNS
<https://tsuname.sidnlabs.nl/>

- **循環参照を意図的に発生させた委任情報を処理させ、フルリゾルバーに委任情報のドメイン名の権威DNSサーバーを攻撃させる**

ループさせる **サイバー攻撃させる**

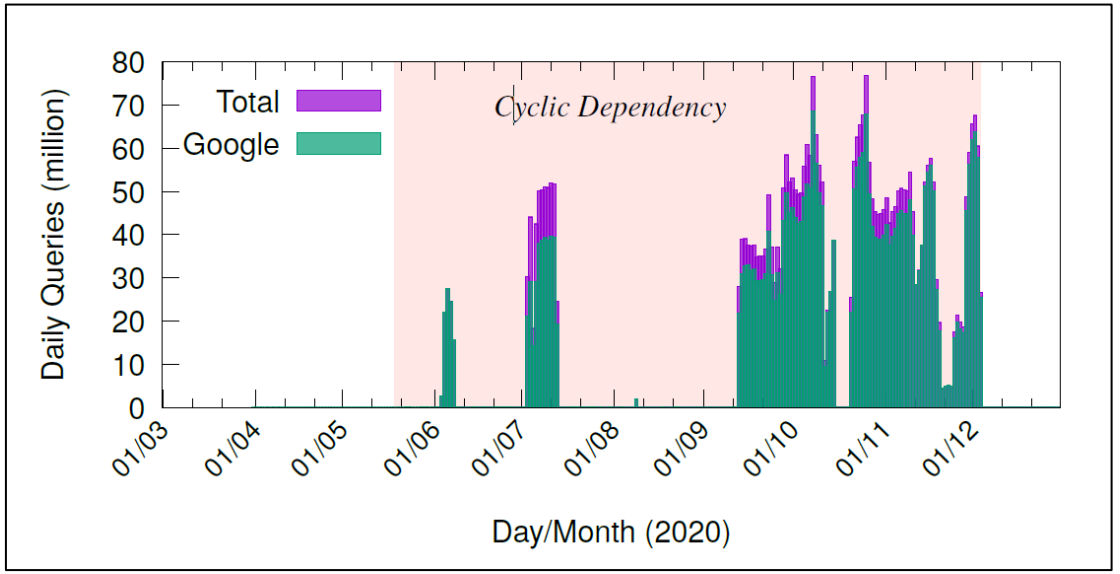
TsuNAMEの仕組み

- 右図のような委任情報の**循環参照**を準備する
- www.example.jp、または www.example.comを名前解決させ、**委任を無限にたどらせる**
- **Google Public DNSが脆弱であった**ため、大量のクエリが発生
  - 現在は対策済み

ループを回避する　**攻撃の効率を下げる**

（jpゾーンに登録）
```
example.jp. IN NS ns1.example.com.
```

（comゾーンに登録）
```
example.com. IN NS ns1.example.jp.
```



引用元：<https://tsuname.sidnlabs.nl/tech_report.pdf>

KeyTrap（CVE-2023-50387）

- 2024年に報告された脆弱性
  - 複数のフルリゾルバー実装・ライブラリ・サービスが脆弱性の対象

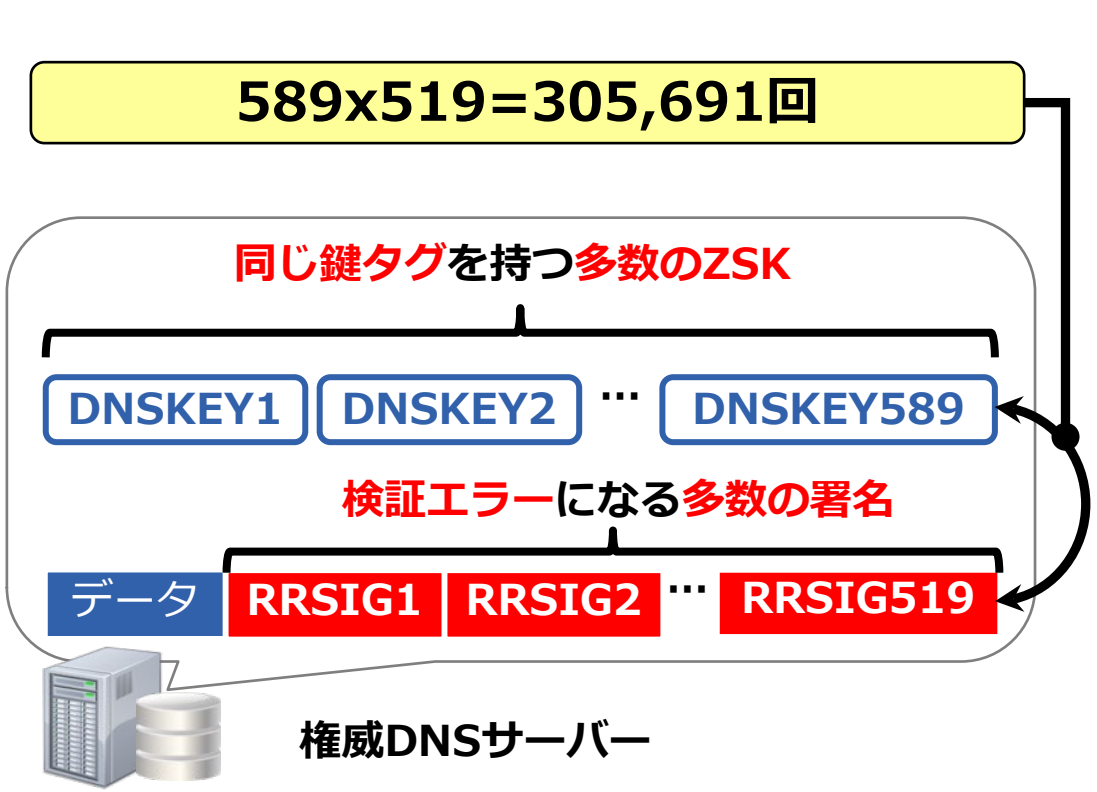
KeyTrap: Serious Vulnerability in the Internet Infrastructure
<https://www.athene-center.de/en/keytrap>

- **負荷の高いDNSSEC検証をさせ、フルリゾルバーを過負荷にする**

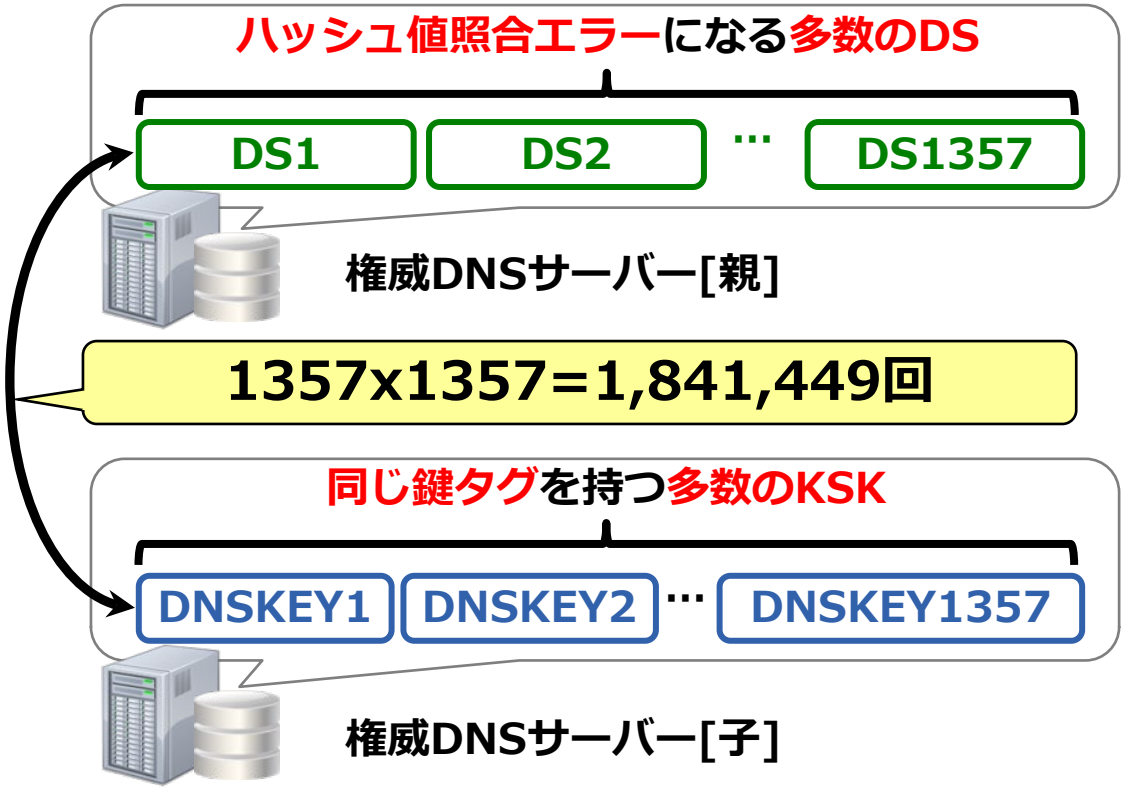
過負荷にさせる

KeyTrapの仕組み

- フルリゾルバーに**負荷の高いDNSSEC検証処理**を強制させる



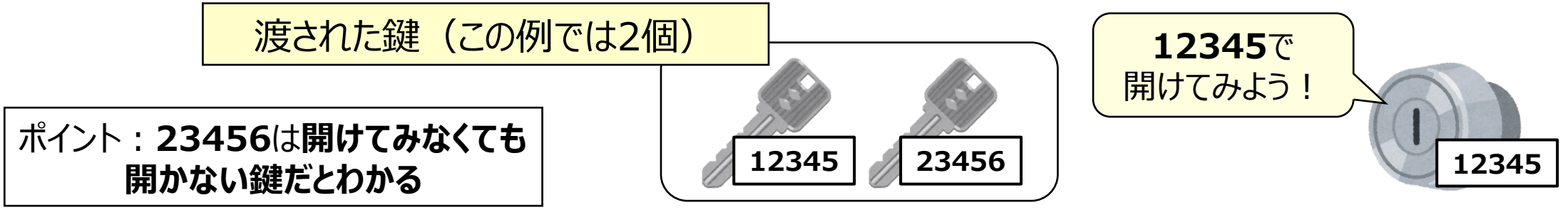
例1：**鍵タグを意図的に衝突させた多数の鍵と検証エラーになる多数の署名**



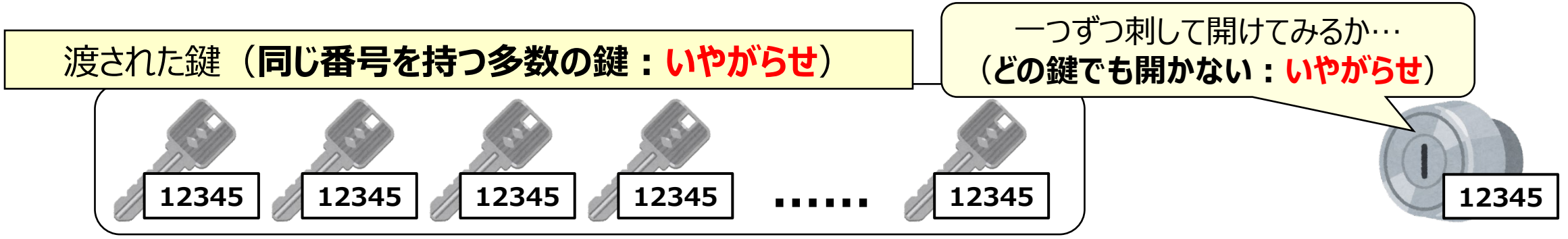
例2：**鍵タグを意図的に衝突させた多数の鍵とハッシュ値照合エラーになる多数のDS**

KeyTrapの攻撃イメージ

- DNSSECでは**鍵と署名の鍵タグを照合**して、検証を効率化している



- **鍵タグを衝突させた多数の鍵を設定し、検証させる**ことで、
  フルリゾルバーに負荷の高いDNSSEC署名検証処理を強制させる



DNSSECの仕様を利用

- DNSSECの仕様では「**署名が複数付加されていた場合、すべての署名検証に失敗した場合のみ検証エラーと判定すること**」と定められている
- 鍵タグを意図的に衝突させた**多数の鍵**と検証エラーになる**多数の署名**で、**フルリゾルバーをだます**

KeyTrapの対策

- フルリゾルバーの**各実装で対策**　**負荷を下げる**
  - DNSSEC検証の**作業量を抑制する上限値・仕組み**を導入
- さまざまな実装が脆弱であったため、2023年11月から2024年2月にかけ、**脆弱性の報告者とフルリゾルバーの実装者が連携して対応**

種類	脆弱だった実装・サービス
フルリゾルバー	Akamai CacheServe、BIND 9、Knot Resolver、PowerDNS Recursor、Unbound、Windows DNS Server
パブリックDNSサービス	1.1.1.1、Google Public DNS、Quad9、OpenDNS
DNSツール	delv、DNSViz、ldns-verify-zone、kzonecheck
DNSライブラリ	dnspython、getdns、ldns、libunbound

このパートのまとめ

- 紹介・解説した脆弱性を、攻撃手法と上限値による対策に注目してまとめると、以下のようになる

脆弱性	攻撃手法	上限値による対策
iDNS Attack	過負荷にさせる / サイバー攻撃させる	負荷を下げる / 攻撃の効率を下げる
NXNSAttack	過負荷にさせる / サイバー攻撃させる	負荷を下げる / 攻撃の効率を下げる
TsuNAME	ループさせる / サイバー攻撃させる	ループを回避する / 攻撃の効率を下げる
KeyTrap	過負荷にさせる	負荷を下げる

- 次のパートではここまでの説明を踏まえ、ちょうどいいDNSのために必要なことについて考える

3. ちょうどいいDNSのために必要なこと

ここからは**森下と熊谷の掛け合い**で進めます。

フルリゾルバーは頑張り過ぎ？

- 前のパートを聞いていて思ったんですが、紹介した攻撃手法はどれも、**頑張り過ぎるフルリゾルバーが原因になってる**んじゃないでしょうか？
- 人間もフルリゾルバーも、**頑張り過ぎは良くない**んじゃないでしょうか？
- **KeyTrapの論文タイトル**にも、そう書いてありました。

The Harder You Try, The Harder You Fail:
The KeyTrap Denial-of-Service Algorithmic Complexity Attacks on DNSSEC
<https://arxiv.org/abs/2406.03133>

The Harder You Try, The Harder You Fail：頑張れば頑張るほど失敗する

フルリゾルバーにとって一番大事なこと

- そうだね。実は**フルリゾルバーにとって一番大事なこと**は何か、ということは、**RFC 1034にちゃんと書いてあった**んだ。

> The recommended priorities for the resolver designer are:
>
> 1. **Bound the amount of work (packets sent, parallel processes started)** so that a request can't get into an infinite loop or start off a chain reaction of requests or queries with other implementations EVEN IF SOMEONE HAS INCORRECTLY CONFIGURED SOME DATA.

> リゾルバー設計者に推奨される優先順位は以下の通りである：
>
> 1. 「たとえ誰かがあるデータを誤って設定していたとしても」、リクエストが無限ループに陥ったり、他の実装との間にリクエストや問い合わせの連鎖反応が引き起こされたりしないように、**作業量（パケット送信数、並列プロセス起動数）を抑制する。**

- みんなつい、**これを忘れて頑張り過ぎちゃう**んだね。

頑張り過ぎの背景にあること

- 頑張り過ぎちゃう背景には、インターネットで伝統的に尊ばれてきた**ポステルの法則**と、**名前解決の効率を上げたいという実装者の意思**があると思う。
- 実は、ポステルの法則については**過剰な適用に対する見直し**が進められていて、最近IABがとりまとめた**RFC 9413**では、**ロバストネス原則にはソフトウェアの欠陥・サイバー攻撃・予期しない入力に対する耐性も含むので、「受信では寛容に」の過剰な適用は危険**だということになっているんだ。

RFC 9413 - Maintaining Robust Protocols
<https://datatracker.ietf.org/doc/rfc9413/>

RFC 9413（2023年6月発行）

どうすればいいのか？

- じゃあ、これからどうすればいいんですか？
- きっとDNSも人間と同じで「**倒れない程度に頑張る**」のがいいんじゃないかな。つまり、**一番大事なことができなくならないように気を付けながら、できる範囲で頑張る**と。結構難しいけどね。
- 難しそうですね。。
- たぶん「**ちょうどいいDNS**」を目指すのがいいんじゃないかと思う。
- ちょうどいいDNS、**今回のテーマ**ですね！

「ちょうどいい」とは

- 「**ちょうど**」を辞書で引いてみると、こんなことが書いてある。
  - ① ある基準に、**過不足なく一致する**さま。きっかり。ぴったり。きっちり。
    - 「―約束の時間に着く」「ブラジルは―日本の裏側にある」
  - ② ある物事が**期待・目的にうまく合う**さま。折よく。都合よく。
    - 「―よいところへ来てくれた」「―手があいたところだ」

出典：デジタル大辞泉（小学館）

- つまりちょうどいいは「**過不足なく一致し、期待・目的にうまく合うこと**」ということになる。
- なるほど！

「ちょうどいいDNS」とは

- 過去のランチセミナーでも話したように、DNSは**プロトコル・実装・運用**の三つが重要だ。だから、ちょうどいいDNSを考える場合も、プロトコル・実装・運用の**三つの観点で考える**必要がある。
- そうですね。「プロトコル・実装・運用」の話は入社してから今まで、いろいろなところで聞きました。
- ここから、**ちょうどいいDNSを実現するために進められている活動**の例について、プロトコル・実装・運用の三つの観点で紹介していこう。

プロトコルにおける活動（1/2）

- 今、**IETF**にこんな提案が出されている。

> Upper limit values for DNS
> <https://datatracker.ietf.org/doc/draft-fujiwara-dnsop-dns-upper-limit-values/>

- この提案はDNSを丈夫にするために、**DNSプロトコルのさまざまな値について、合理的な上限値を設定しよう**というものなんだ。
  ここまでに話した内容も含まれている。
- 提案者はこの人ね。
  - draft-**fujiwara**-dnsop-dns-upper-limit-values

プロトコルにおける活動（2/2）

- さっき、「**最大積載量：積めるだけ**」という話をしたけど、この提案は事故や妨害行為が起こりにくくなるように、**荷物の種類や性質などを考えた上で、安全な最大積載量を決めておこう**、ということだね。
- IETFでの議論はまだ始まったばかりだけど、**この提案はプロトコルの観点から、ちょうどいいDNSを実現するための活動**の一つだと思う。
- この後のDNS DAYで話すはずだから、よく聞いておこう。

実装における活動（1/2）

- 今日紹介したように、**上限値の設定が十分でないことを利用したDNSの脆弱性**が、これまでに何度か公開されている。
- KeyTrap脆弱性が公開された時、ISCでBIND 9の実装を担当しているPetr Špaček氏が**印象的なブログ**を公開していたので、紹介しておこう。

BIND 9 Security Release and Multi-Vendor Vulnerability Handling, CVE-2023-50387 and CVE-2023-50868 – ISC
<https://www.isc.org/blogs/2024-bind-security-release/>

Petr Špaček氏（ISC）の弁明（1/2）

- 2024年2月14日のISCブログから、印象的な部分を引用した。
  太字の部分がその一節。

> Indeed, you read it correctly, even back in 1987 when the original DNS specification was written, the top priority was limiting the amount of work done by the implementation! **Again and again, researchers continue to show implementers the dark corners where this simple instruction was not followed.**

- 英語ですね。。。

引用元：<https://www.isc.org/blogs/2024-bind-security-release/>

Petr Špaček氏（ISC）の弁明（2/2）

- 翻訳するとこうなる。**暗黒面（dark corners）**という言葉が印象深い。

> いかにも、1987年にオリジナルのDNS仕様が書かれた時でさえ、実装によって行われた作業量を制限することが最優先事項であったのだ！**何度も何度も、研究者たちは、この単純な指示に従わなかったという暗黒面（dark corners）を、実装者に示し続けている。**

- **実装の敗北**ということですか。。
- そう、つまり実装者は**何度も敗北している**と言っているわけだ。

実装における活動（2/2）

- だとすると実装者はその都度、**フルリゾルバーにとって一番大事なことは何かを思い知らされている**とも言えますね。
- そういうことになるね。でも、現在の実装やサービスに設定されている**上限値や動作の工夫**は、**敗北から学んだ結果**であるとも言える。
- そういう意味で、**実装やサービスが脆弱性やサイバー攻撃で鍛えられて、適切な上限値の設定や動作が定まっていくことは、ちょうどいいDNSを実現するための活動**であると言えるんじゃないかな。
- 何だか深いですね。。

運用における活動（1/2）

- そして、DNSは**複数の構成要素が連携し合って動いている分散システム**だから、うまく動かすには**それぞれのサーバーやシステムを運用している関係者が、互いに連携・協調し合う**必要がある。
- 例えば、**ちょうどいいDNSの設定と運用のためのこれまでの活動**として、2019年と2020年に実施された、**DNS flag day**が挙げられる。

運用における活動（2/2）

- DNS flag dayではEDNS0における**ワークアラウンド処理の削除**や、IPフラグメンテーション回避のための**EDNS0バッファサイズの上限値の導入**を、**日付を決めて一斉に実施する**ことが試みられたんだ。
- ということは、DNSの重要な設定変更や上限値を導入する場合、**DNS flag dayがまた実施されるかも知れない**んですね。
- そういうことになるね。脆弱性対応もそうだけど、**普段から情報共有や情報交換を図ったり、最新動向に気を配っておく**ことが、**ちょうどいいDNSの設定と運用**を実現するために、とても重要になるね。

おわりに：
ちょうどいいDNSのために必要なこと

- ということで、今回のランチセミナーでは**上限値**をキーワードとして、**ちょうどいいDNSを実現するために必要なこと**について考えてみました。
- 安定・柔軟で限界を超えない「ちょうどいいDNS」の実現にも、**DNSに関わるさまざまな立場の関係者がそれぞれの役割を果たし、連携し合って活動を続けていく**ことが、やっぱり重要なんじゃないかなと思います。

プロトコル設計者　**DNSサービス提供者**　**サーバーソフトウェア開発者**

サーバー運用者　**ネットワーク運用者**　**脆弱性の発見者**　**他にも…**

これからも力を合わせて、倒れない程度に頑張っていきましょう

最後までお聞きいただき
ありがとうございました！

<https://jprs.jp/tech/>

アンケートにご回答お願いします！

@JPRS_official　JPRSofficial　JPRSpress