DNSを狙った攻撃の影響範囲と フルリゾルバーの可用性・信頼性を高めるためのポイント

～KeyTrap脆弱性を題材として～

2024年6月12～14日

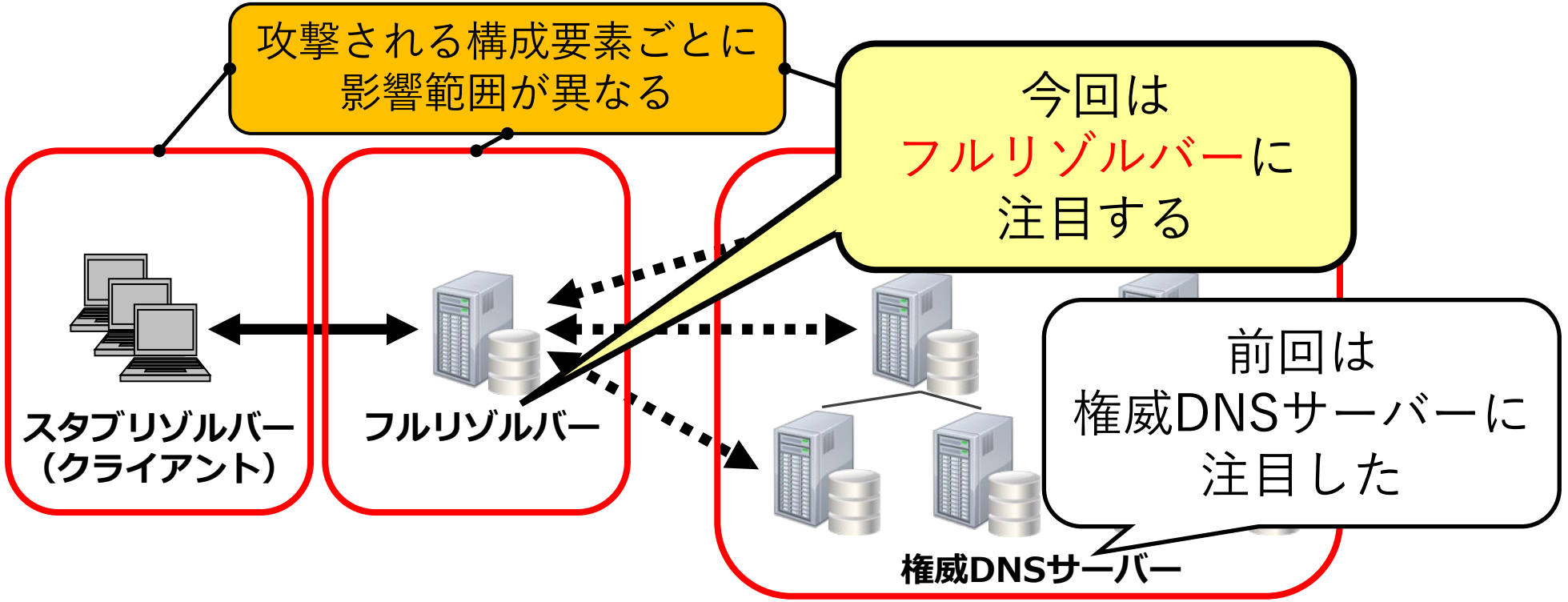
Interop Tokyo 2024

本セミナーの内容

- DNSを狙った攻撃の影響範囲
- KeyTrap脆弱性の概要
- フルリゾルバーの可用性・信頼性を高めるためのポイント
- 運用担当者・責任者のみなさまへ

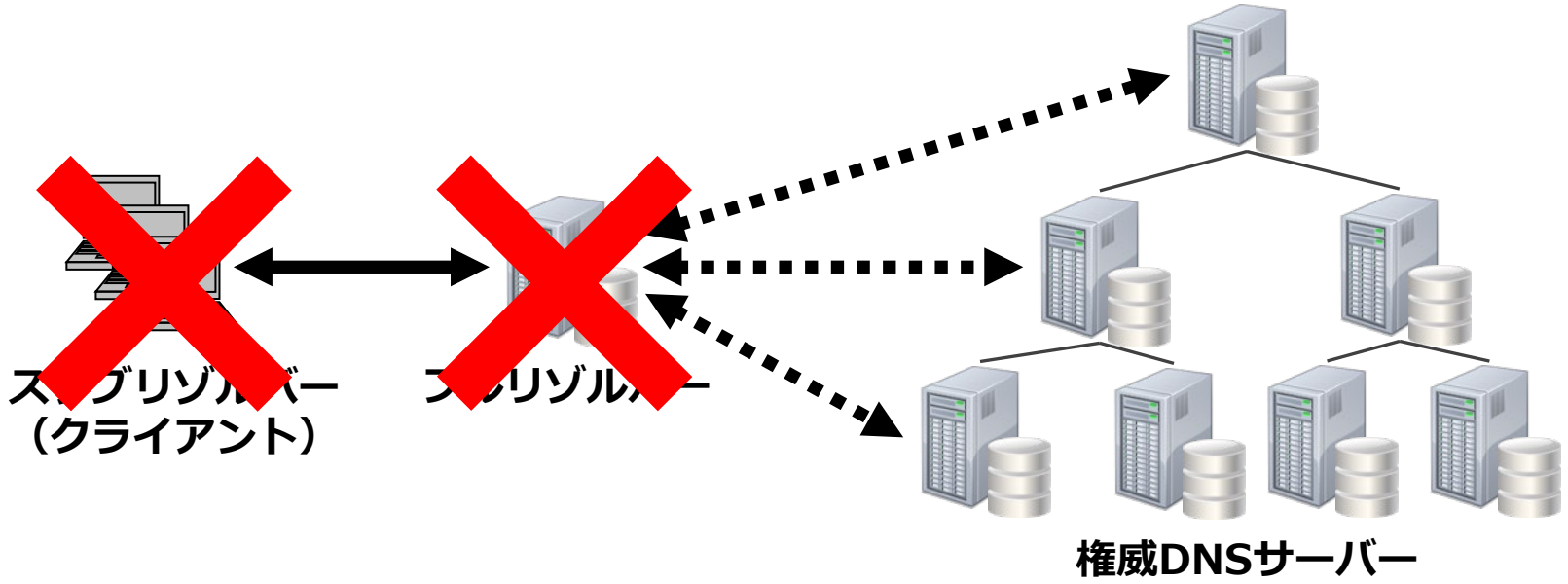
DNSの構成要素と攻撃の影響範囲

- DNSが攻撃されてサービスに影響が及んだ場合、攻撃される構成要素ごとに、その範囲が異なる



フルリゾルバーが攻撃された場合の影響範囲

- フルリゾルバーが攻撃されてサービスに影響が及んだ場合、
  そのフルリゾルバーを使っているすべての利用者に影響が及ぶ
  - 利用者の「インターネットが使えない・おかしい」に直接つながるため、高い可用性・信頼性が要求される



「使えない」と「おかしい」

- 使えない：フルリゾルバーが応答を返さなくなる
  - 例：DDoS攻撃、脆弱性を突いたDoS攻撃など
- おかしい：フルリゾルバーが不適切な応答を返す
  - 例：キャッシュポイズニング、登録情報の不正書き換えによるドメイン名ハイジャックなど

本セミナーでは「インターネットが使えない」に注目する

ここからの説明の流れ

- 2024年2月に「KeyTrap」という、フルリゾルバーを狙う新たな脆弱性が発表された
  - フルリゾルバーを使えなくすることを狙った脆弱性
- 本セミナーではKeyTrap脆弱性を題材として、フルリゾルバーの可用性と信頼性を高めるためのポイントについて解説する

「DNSに対する最悪の攻撃」？

セキュリティニュースアラート

「DNSに対する最悪の攻撃」　DNSSEC設計の根幹に関わる脆弱性「KeyTrap」が見つかる

ATHENEはDNSSECの設計に深刻な欠陥があると発表した。この脆弱性を悪用すると単一のDNSパケットで全てのDNS実装とパブリックDNSプロバイダーを停止状態にすることが可能だという。

2024年02月19日 08時00分 公開　　[後藤大地，有限会社オングス]

印刷　通知　見る　Share　B!　77

ドイツの国立応用サイバーセキュリティ研究センターATHENEは2024年2月13日（現地時間、以下同）、DNSSEC（ドメインネームシステムのセキュリティ拡張機能）の設計に重大な欠陥「KeyTrap」を発見したと発表した。

記事引用元：<https://www.itmedia.co.jp/enterprise/articles/2402/19/news054.html>

KeyTrap脆弱性（CVE-2023-50387）

- サービス不能（DoS）攻撃の新たな手法
  - 2024年2月にドイツの研究グループが公開
  - 共通脆弱性識別子（CVE）として、CVE-2023-50387が割り当て済み
- フルリゾルバーをだまして負荷の高い処理をさせるように仕向け、サービス不能の状態にする

負荷の高い処理をさせる方法

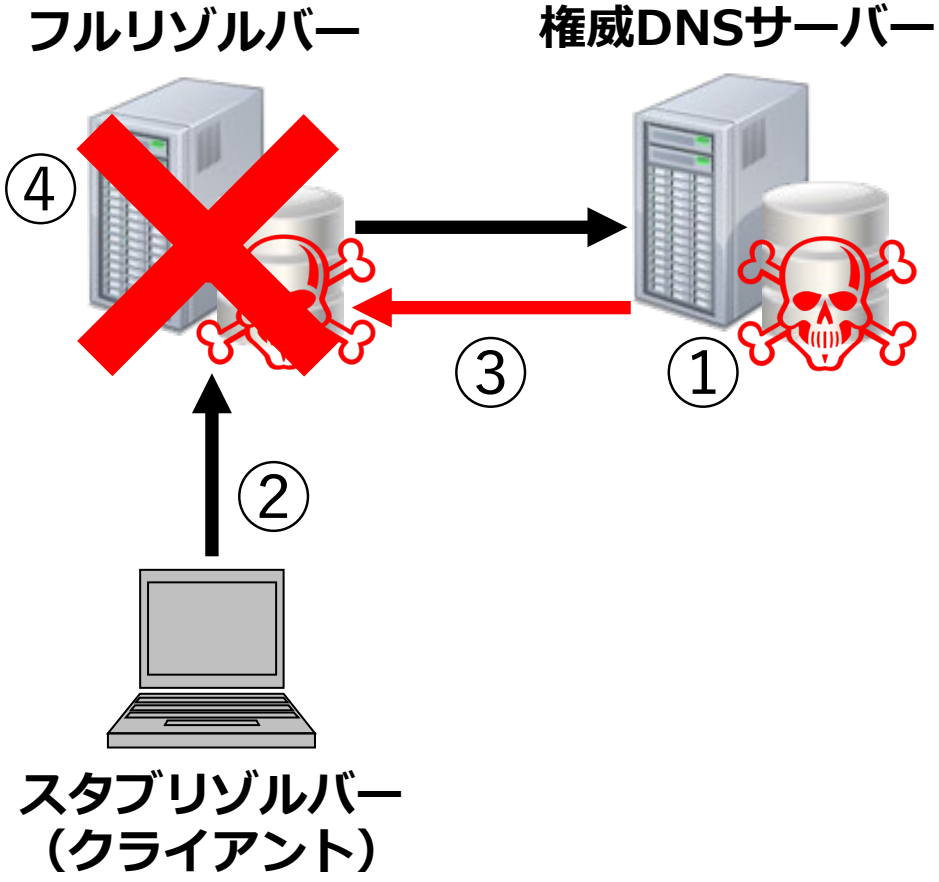
- DNSの名前解決を利用して、攻撃用のデータを注入する

① 攻撃者が制御可能な権威DNSサーバーに、多大な作業を要するデータを設定する

② そのデータを名前解決させる

③ 権威DNSサーバーからデータが注入される

④ フルリゾルバーが過負荷に陥る

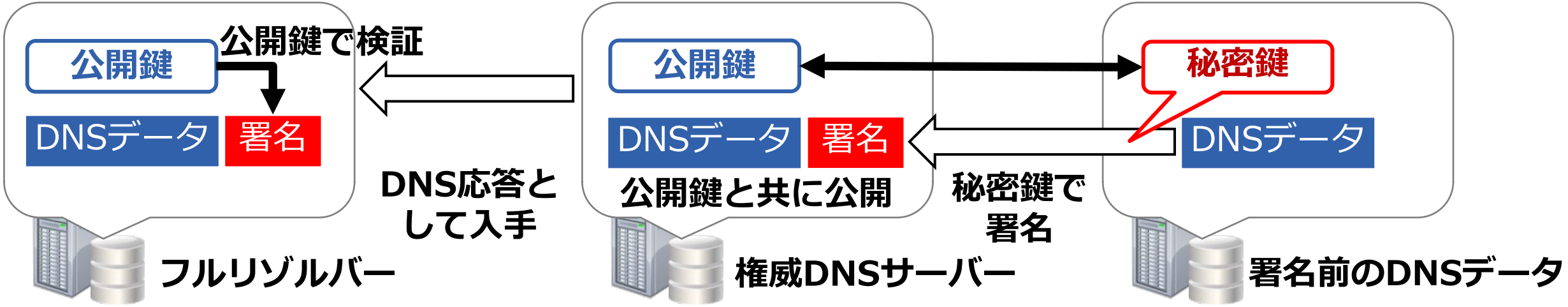


DNSSEC検証の仕組みを利用

- フルリゾルバーに負荷の高い処理をさせるため、DNSSECの鍵と署名検証の仕組みを利用している
- 「KeyTrap」という名前の由来になっている
  - 鍵（Key）と署名検証の仕組みで罠（Trap）にはめる

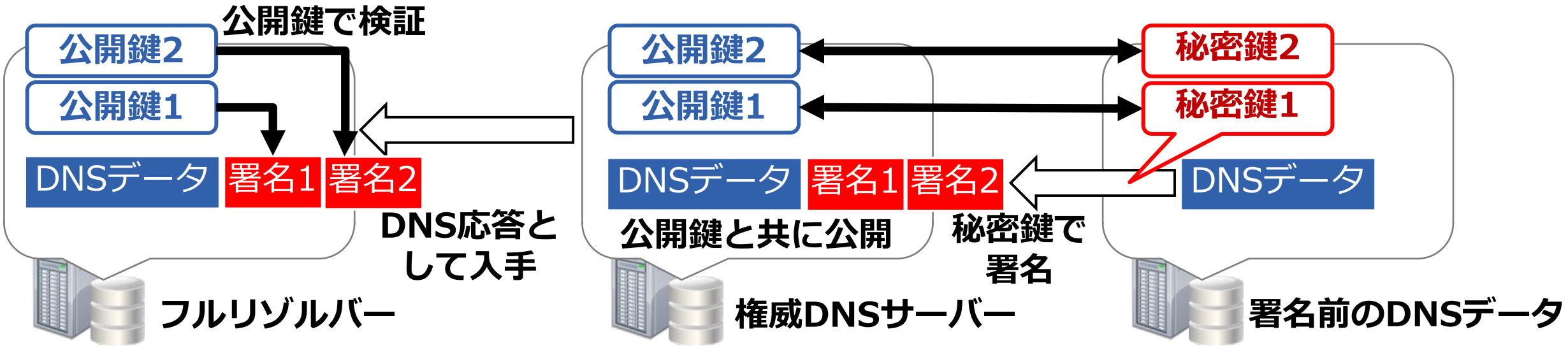
DNSSECの鍵と署名検証

- 通常の運用では、DNSSECの鍵・署名は1セット
  - DNSデータを秘密鍵で署名し、署名付きのDNSデータを公開鍵と共に公開して、フルリゾルバーで署名を検証



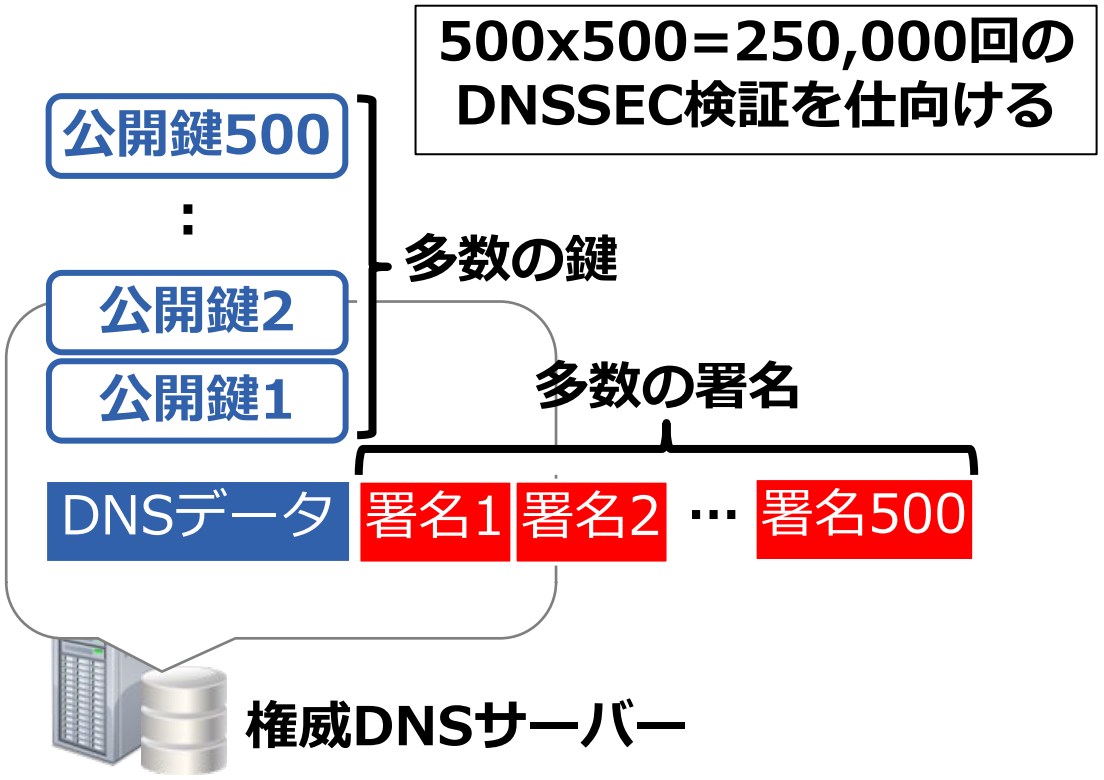
複数の鍵・署名の設定

- 鍵の更新や運用者の変更を容易にするため、DNSSECでは複数の鍵・署名を設定できるようになっている
  - 複数の鍵を設定でき、それぞれの鍵で署名を検証できる



KeyTrap脆弱性の仕組み

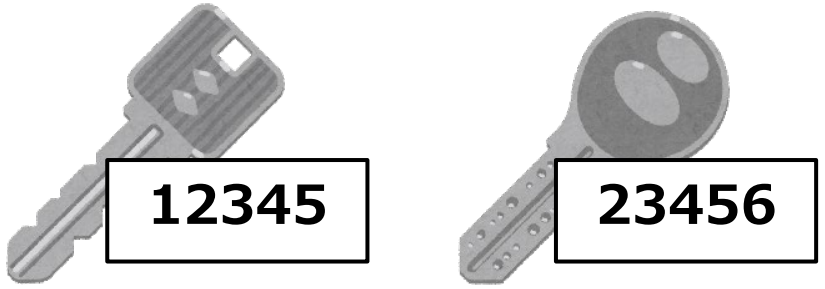
- この仕組みを利用し、多数の鍵と多数の署名を持つDNSデータを作成・公開して、フルリゾルバーにそれらすべてをDNSSEC検証させるように仕向ける



なぜ、そんな単純な方法で攻撃できたのか？

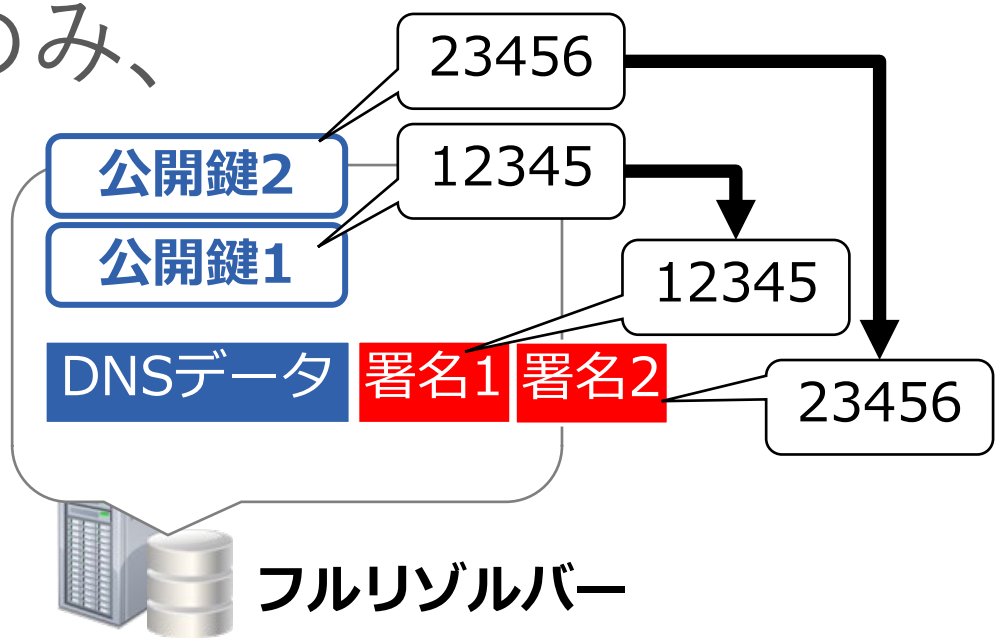
攻撃者の工夫①：鍵タグの衝突（1/3）

- DNSSECには、検証すべき鍵・署名を見分ける、「鍵タグ」と呼ばれる仕組みがある
- 鍵タグとは？
  - 鍵を識別するために、鍵ごとに定められる番号
    - 鍵から計算した16ビットの数値



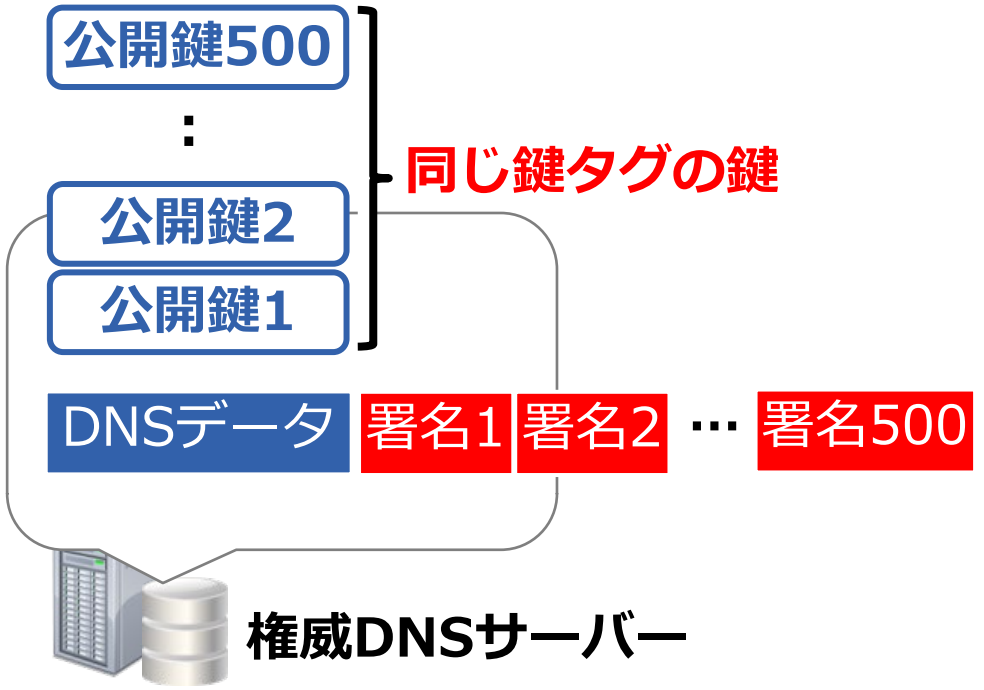
攻撃者の工夫①：鍵タグの衝突（2/3）

- どの鍵で署名したかがわかるように、署名には使った鍵の鍵タグが埋め込まれるようになっている
- 右図の例では署名1は公開鍵1のみ、署名2は公開鍵2のみで検証できるようにすることで、署名検証の負荷を減らしている



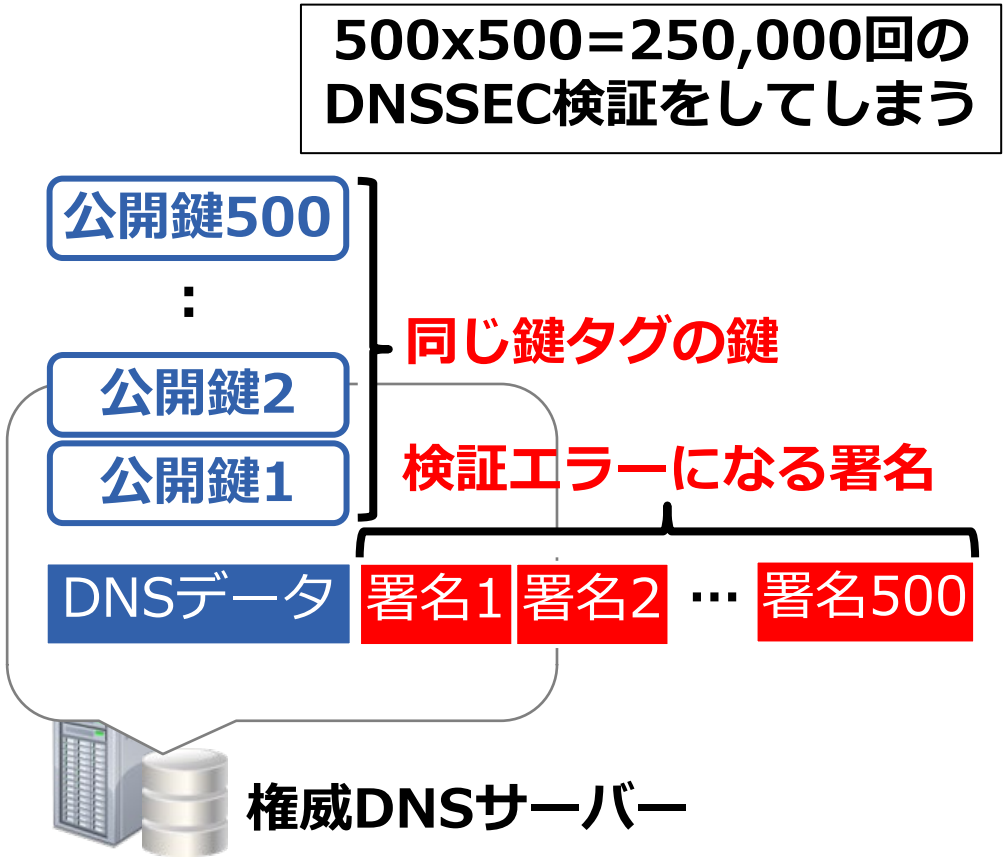
攻撃者の工夫①：鍵タグの衝突（3/3）

- 鍵タグを意図的に衝突させた多数の鍵を作成・設定して、この仕組みをバイパスさせる
  - 鍵タグは公開鍵（DNSKEYレコード）のチェックサムであるため、鍵を作り続けることで同じ鍵タグを持つ鍵を作成可能



攻撃者の工夫②：検証エラーになる署名

- その上で、すべての署名を検証させるため、検証エラーになる多数の署名を意図的に付加する
  - DNSSEC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



同じ鍵タグの鍵と検証エラーになる署名で、フルリゾルバーをだます

KeyTrap脆弱性の影響（1/2）

- 複数の実装・サービスが脆弱であった
  - 複数のフルリゾルバー
  - 複数のパブリックDNSサービス
  - 複数のDNSツール・DNSライブラリ

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

KeyTrap脆弱性の影響（2/2）

- 影響の程度は実装によりさまざまであった
- 参考：一つのDNS応答で実験環境をDoSできた時間
  - Akamai CacheServe/PowerDNS Recursor/Stubby：約3分
  - Unbound：1014秒（約17分）
  - BIND 9：16時間以上
    - 実装の効率が悪く、検証ごとに他のすべての鍵も解析し直していた
  - Knot Resolver：56秒
    - ゾーンごとの鍵数に上限が設けられており、DoS継続時間が短かかった

この「16時間以上」が
センセーショナルに報道された

KeyTrap脆弱性の対策

- フルリゾルバーの各実装で対策された
  - DNSSEC検証の作業量を抑制する仕組みを導入
- RFC 1034に記述されている、フルリゾルバーの最優先事項を実装

> 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。

引用元：RFC 1034日本語訳：<https://jprs.jp/tech/material/rfc/RFC1034-ja.txt>

悪意を持つデータにだまされないように、各実装で対策

フルリゾルバーの可用性・信頼性を高めるためのポイント

可用性・信頼性を高めるためには…

- フルリゾルバーの機能を守り、適切に動作させることが重要
  - 「使えない」を起こさず、起こってもすぐに直せるように運用する
- そのためには、障害の予防と速やかな対応が必要
  - 予防：障害の発生を防ぐ
  - 速やかな対応：障害を見つけ、すぐに直す
- そのためには、設計・構築と運用の双方における考慮が必要
  - 設計・構築：作る時のポイント
  - 運用：動かす時のポイント

以降ではフルリゾルバーにおける
これらのポイントについて解説する

障害の発生を防ぐ（1/2）

作る時のポイント

- フルリゾルバーの設計・構築におけるポイントの例
  - サーバーの可用性・信頼性の向上
    - サーバーの多重化・DNSソフトウェアの多様化、専用のアプライアンスサーバーの導入、など
      - 運用しやすいソフトウェア・システムを選ぶことも重要
  - 十分なネットワーク帯域・アクセシビリティの確保
    - 組織内と組織外の双方への配慮が必要
      - フルリゾルバーは組織内からアクセスされ、組織外にアクセスする

障害の発生を防ぐ（2/2）

動かす時のポイント

- フルリゾルバーの運用におけるポイントの例
  - 脆弱性やサイバー攻撃などの脅威への対応
    - 脆弱性情報・運用情報・サイバー攻撃動向の調査・把握
    - 適切なバージョンアップ・パッチの適用

重大な脆弱性では関係者間で連携が図られ、脆弱性情報とパッチが同時に公開されることが多い（KeyTrapもその形で対応）

  - 攻撃に晒される・踏み台にされるリスクの低減
    - 提供範囲外の利用者からのアクセスを制限、など

障害を見つけてすぐに直す（1/2）

作る時のポイント

- フルリゾルバーの設計・構築におけるポイントの例
  - 権威DNSサーバーとフルリゾルバーの分離
    - 機能の相互干渉を防ぐ
    - 障害の切り分け・対応を容易にする
  - 障害の発生を想定した設計・構築の実施
    - 切り分け・切り離し・復旧が容易なシステムの設計・構築
      - 名前解決サービスに影響を与えずに実行できることが望ましい

障害を見つけてすぐに直す（2/2）

動かす時のポイント

- フルリゾルバーの運用におけるポイントの例
  - 運用状況の監視
    - 名前解決の状況（可否、時間）
    - リソースの使用状況（CPU・メモリ・トラフィックなど）
  - 統計情報・ログの取得・分析
    - 単位時間やリソースレコードごとのクエリ数・レスポンス数など
  - 障害発生時の運用体制の構築・訓練の実施
    - 障害の発生・対応に普段から備えておく

運用担当者・責任者のみなさまへ

フルリゾルバーも守りましょう！

- フルリゾルバーはネットワークの利用と運用において必須の構成要素であり、高い可用性・信頼性が求められる
  - にも関わらず、そのことはあまり意識されていない
- フルリゾルバーの可用性・信頼性を高めるためには、その特性に由来するポイントを意識した設計・構築・運用が必要になる

常日頃からフルリゾルバーに気を配り、守ることが、
ネットワークの利用と運用の安定につながる

権威DNSサーバーと共に、フルリゾルバーも守りましょう！