未熟なDNSと今後どう付き合うべきか

委任／移転通知インジェクション攻撃と DNS Water Torture（Slow Drip）攻撃~~について考える~~

を題材として

2014年11月20日

Internet Week 2014 ランチセミナー

株式会社日本レジストリサービス（JPRS）

森下 泰宏・久保田 秀

講師自己紹介

- 森下 泰宏（もりした やすひろ）
  - 日本レジストリサービス（JPRS） 広報宣伝室
  - 主な業務内容：技術広報担当として、ドメイン名・DNSに関する技術情報を分かりやすく伝える
  - 最近思うこと：今年はもうお腹いっぱいです・・・
- 久保田 秀（くぼた しゅう）
  - 日本レジストリサービス（JPRS） システム部
  - 主な業務内容：レジストリシステム、gTLD取次システム及びその周辺システムの開発と運用
  - 最近思うこと：「重複」のない平穏な年末を・・・

本日の内容

1. 委任／移転通知インジェクション攻撃の概要と対策
2. DNS Water Torture（Slow Drip）攻撃の概要と対策
   - 本セミナーでは以降「DNS水責め攻撃」と呼びます
3. 未熟なDNSと今後どう付き合うべきか

1と3を森下が、2を久保田が担当します

本日の目標

- DNSの基本部分の弱点を狙う二つの攻撃手法の概要と対策の現状について解説し、
- それを題材として、そうした「未熟なDNS」と今後どう付き合っていくかについて、みんなで考えていくためのきっかけとする

  ・・・ことを目指します

注：今日、問題が全て解決するわけでも、
問題の解決を諦めるわけでもありません

1. 委任／移転通知インジェクション攻撃の概要と対策

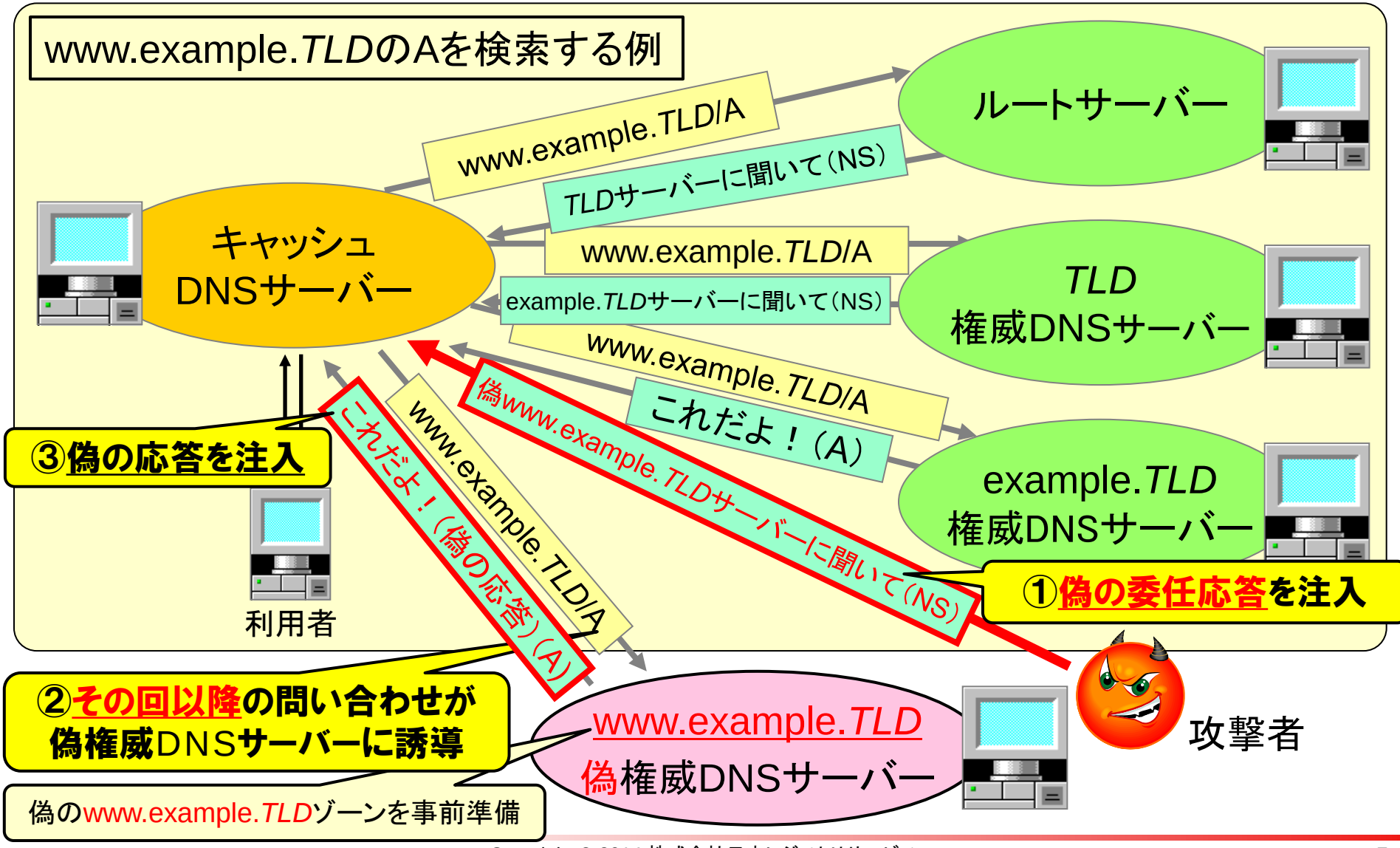
このパートの構成

- 攻撃の原理
- 共通点と相違点
- 攻撃が成立する理由
- 攻撃対象となりうるドメイン名
- IETFにおける議論状況
- 標準化・提案・公開された文書の紹介

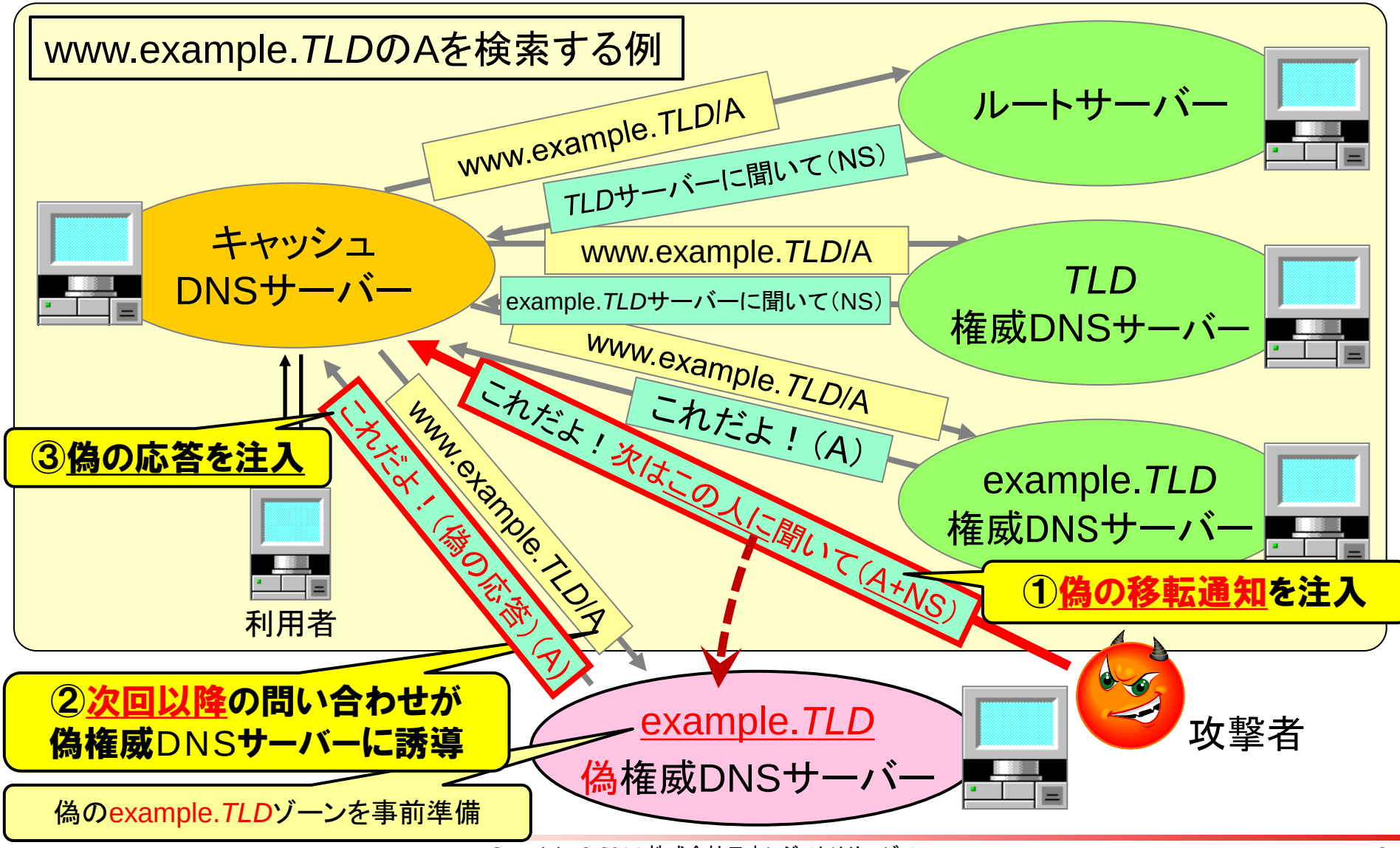
委任インジェクションと移転通知インジェクションという、
よく似た2種類の異なる攻撃手法を比較しながら解説

混乱・混同しないように注意

委任インジェクション攻撃の原理



移転通知インジェクション攻撃の原理

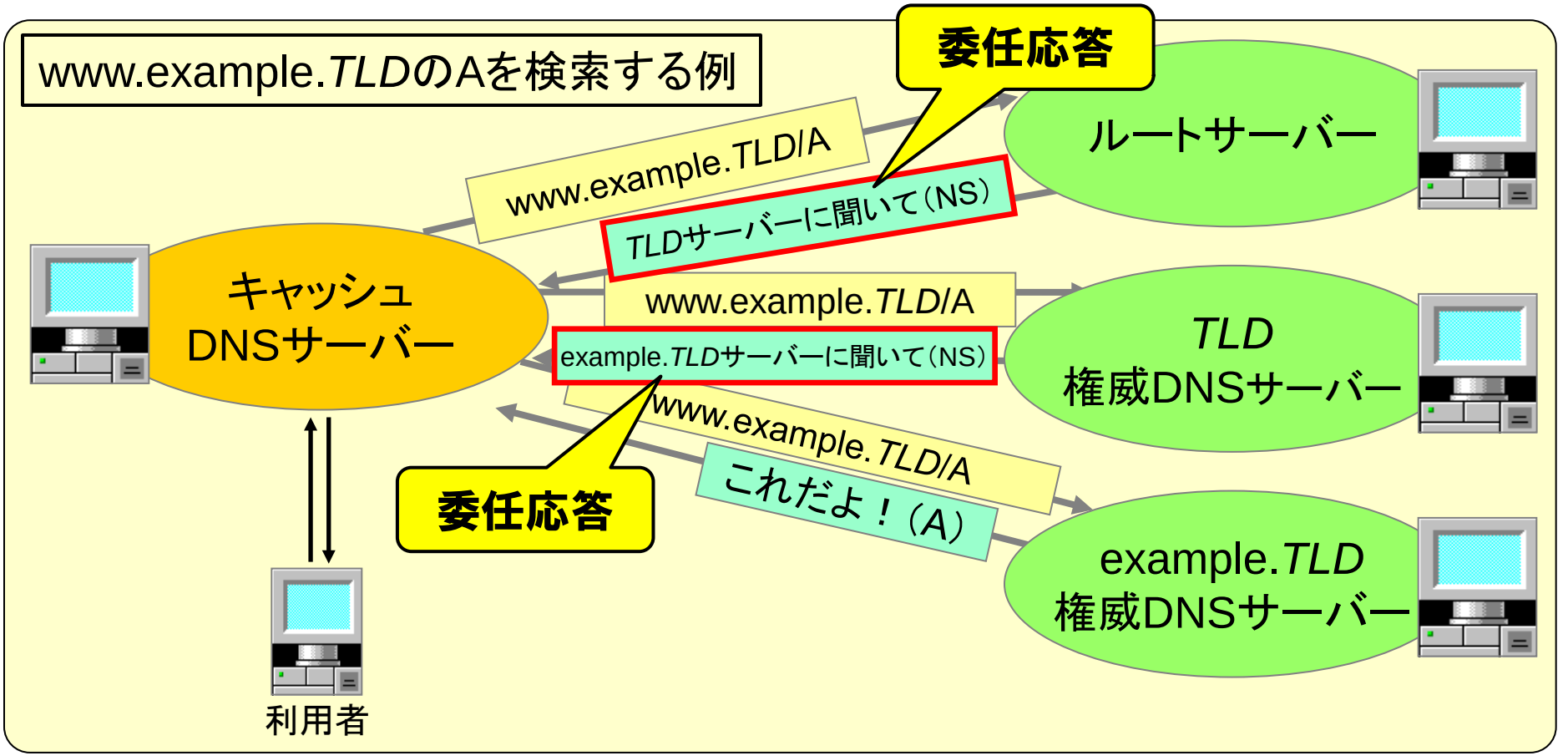


共通点と相違点

- 共通点：攻撃対象がNSレコードである
  - 偽のNSレコードをキャッシュに注入し、
    名前解決を偽の権威DNSサーバーに誘導する
- 相違点：注入する応答の種類が異なる
  - 委任インジェクション攻撃
    - 偽の委任応答によりNSレコードを注入する
  - 移転通知インジェクション攻撃
    - 偽の移転通知によりNSレコードを注入する

委任応答／移転通知とはそれぞれ何か？

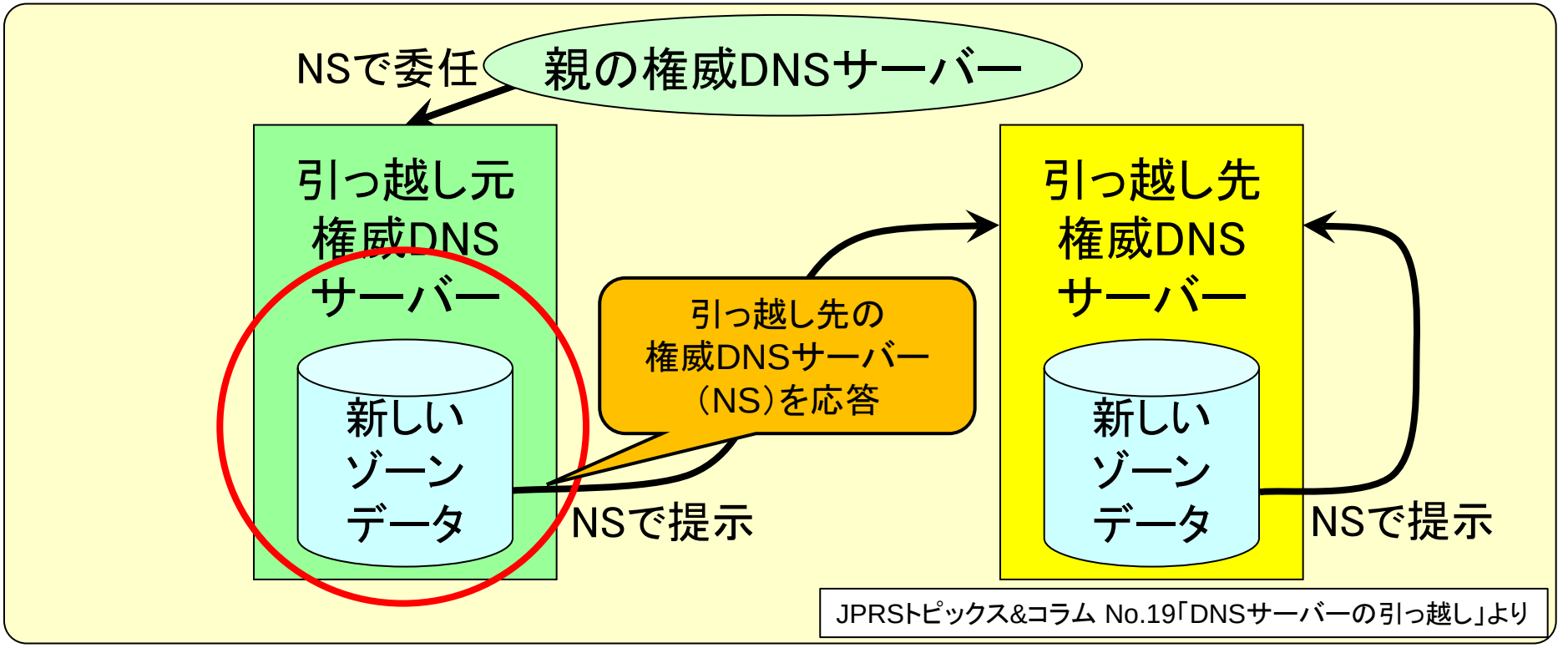
委任応答とは？



委任先の権威DNSサーバー（NS）を伝える応答
名前空間の分散管理実現のために必須

移転通知とは？

- 権威DNSサーバーの引っ越し作業の途中で出現



次回以降の問い合わせで使ってほしい引っ越し先の
権威DNSサーバー（NS）を、通常応答に付加して応答

攻撃成立の理由

- いずれも、DNSの仕様上の弱点により攻撃が成立
- 委任インジェクション攻撃
  - ゾーン頂点（委任の境界）が委任先（子）の管理になる
    - DNSの基本仕様であり、全ての実装が影響を受ける
- 移転通知インジェクション攻撃
  - 移転通知（子のNS）は委任応答（親のNS）よりも信頼度（trustworthiness）が上
    - RFC 2181 5.4.1. Ranking data
  - 上記仕様に準拠しない実装では攻撃は成立しない
    - Vantio CacheServeやMaraDNS/Deadwoodなど
    - 副作用の例：権威DNSサーバーの引っ越しに時間がかかる

攻撃対象となりうるドメイン名（委任インジェクション攻撃）（1/2）

- “.”以外の全ての名前が攻撃対象となりうる
  - 委任されうる全ての名前
- ただし、攻撃対象ドメイン名のNSレコードがキャッシュされている場合、その名前は攻撃できない
  - 委任応答を外部から注入できない
- 例：example.*TLD*のNSがキャッシュされていると、
  - example.*TLD*は委任インジェクション攻撃できない
  - www.example.*TLD*は委任インジェクション攻撃できる

攻撃対象となりうるドメイン名（委任インジェクション攻撃）（2/2）

- そのため、攻撃対象となりうるものは以下の二つ

1. NSレコードが存在しないドメイン名
   - 通常（末端）のドメイン名やネームサーバーホスト名
     - 例：一般的なホスト名やネームサーバーホスト名全て
   - 管理上の理由でNSレコードが存在しないドメイン名
     - 例：属性型JPドメイン名の第2レベル、LG.JPドメイン名の第2レベルや第3レベル、など
2. NSレコードがキャッシュされにくいドメイン名
   - いわゆる親子同居している場合の子側
     - 次ページの例を参照

「親子同居の子側」の例

例：*TLD*とxx.*TLD*が親子同居の状態

```
$ORIGIN TLD. ; 親（TLD）ゾーン
@  IN SOA …
   IN NS a.dns.TLD.
xx IN NS a.dns.TLD.

$ORIGIN xx.TLD. ; 子（xx.TLD）ゾーン
@       IN SOA …
        IN NS a.dns.TLD.
example IN NS ns.example.xx.TLD.
```

← このNSレコードが応答されない ⇒ キャッシュされない

← このNSレコードが応答される

- 上記の設定がされている（*TLD*とxx.*TLD*が親子同居）権威DNSサーバーにwww.example.xx.*TLD*/Aを問い合わせた場合、xx.*TLD*に対する委任応答（NSレコード）が返らない
- つまり、xx.*TLD*のNSレコードが存在しない場合と同じ状況
  - このようなドメイン名は世界中に存在（属性を持つccTLDなど）

攻撃対象となりうるドメイン名（移転通知インジェクション攻撃）

- “.”を含む全ての名前が攻撃対象となりうる
  - 実際には“.”は移転通知インジェクション攻撃できない
    - BIND 9やUnboundは起動時にプライミングを実施している
      - プライミングで得たNSレコードは移転通知よりも信頼度が高い
- そのゾーンに一般利用者から参照される名前（= 通常応答の対象となる名前）が存在しない場合、攻撃の対象となりやすい
  - 信頼度が高い（子からの）NSがキャッシュされにくい
    - 基本的に委任のみのドメイン名（ゾーン）
      - 例：全てのTLD（jp、com、netなど）

IETFにおける議論状況

- 委任インジェクション攻撃：2008年8月に発表されたBernhard Müller（ベルンハルト・ミュラー）氏の技術文書が初出
  - Improved DNS spoofing using node re-delegation
    <https://www.sec-consult.com/fxdata/seccons/prod/downloads/whitepaper-dns-node-redelegation.pdf>
- IETFでは2008～2009年に集中的に議論された
  - その結果がRFCとI-Dにまとめられている（後で紹介）
    - RFC 5452
    - draft-barwood-dnsext-fr-resolver-mitigations
    - draft-wijngaards-dnsext-resolver-side-mitigation

IETF関係者の雰囲気

- そのためIETFでは、既に議論済の話題として扱われている模様
  - 最近のdns-operations MLやIETF Meetingなどで、議論が盛り上がらなかった理由の一つと考えられる
- 提案済の対策を粛々と実施すべき
  - ソースポートランダマイゼーション、サーバーの監視
  - 必要に応じた追加対策の適用
  - DNSSECの適用

RFC・I-D・公開資料の紹介

- キャッシュポイズニング攻撃に対する耐性強化・影響緩和策について考察、まとめた文書
- 本日紹介する資料
  - RFC 5452
  - draft-barwood-dnsext-fr-resolver-mitigations
  - draft-wijngaards-dnsext-resolver-side-mitigation
  - Googleの公開資料
  - JPRSの公開資料

RFC 5452

- “Measures for Making DNS More Resilient against Forged Answers”
  - 2009年1月発行、Proposed Standard
  - 目的：偽の応答に対する耐性強化策の標準化
- 実装済の各対策を追認する形で標準化
  - 今となっては当たり前の内容がほとんど
- 記載されている主な項目（抜粋）
  - ID、IPアドレス、ポート番号の取り扱い
  - 権威を持つ応答、持たない応答の取り扱い
  - 内部名のみの受け入れ（注：MUSTではない）
  - ID・ソースポートランダマイゼーションの必須化、など

draft-barwood-dnsext-fr-resolver-mitigations（1/2）

- “Resolver side mitigations”
  - 2008年9月発表、同年10月に-08に更新（未RFC化）
- キャッシュDNSサーバーにおいて取りうる緩和策が評価・検討されている
  - さまざまな緩和策が記述されており、参考になる
    - 一部はGoogle Public DNSなどで実装済
- 参考：このI-Dの著者が開発した、オープンソースのキャッシュDNSサーバーの実装がある模様
  - GbDns <http://sourceforge.net/projects/gbdns/>

draft-barwood-dnsext-fr-resolver-mitigations（2/2）

- Introductionに、カミンスキー型攻撃手法でcomを委任インジェクション攻撃する例が記述されている

> The Kaminsky attack proceeds by asking a recursive DNS server a series of questions, each with a different random prefix, and then sending spoof packets to the server, containing additional records with genuine owner names but invalid data. For example:
>
> Query:
> Question <nonce>.com A
>
> Spoof response:
> Question <nonce>.com A
> Authority: com NS ns.evil.com
>
> The effect is to inject an invalid record into the cache.

draft-wijngaards-dnsext-resolver-side-mitigation

- “Resolver side mitigations”
  - 2009年2月発表、同年8月に-01に更新（未RFC化）
- Unboundの開発者による緩和策の評価・検討
- 書かれている緩和策
  - エントロピーの追加
  - キャッシュにおける注意深い取り扱い
  - 権威を持つデータの再取得
  - 検知と防御
- Unboundにおいて実装済（一部はオプション機能）

Googleの公開資料

- “Security Benefits”
  - <https://developers.google.com/speed/public-dns/docs/security>
- Google Public DNSにおけるセキュリティ施策
- 主な項目（キャッシュポイズニング対策関連）
  - DNSSEC検証
  - 応答の有効性チェック
  - エントロピーの追加
    - 問い合わせ名のランダム化やnonce labelsの付加など
  - 重複する問い合わせの削除（誕生日攻撃対策）

JPRSの公開資料

- 基本対策編（公開済）
  - JPRS DNS関連技術情報 <http://jprs.jp/tech/> に掲載
  - キャッシュDNSサーバー運用者向け
    - ソースポートランダマイゼーションと攻撃の検知・対応を推奨
  - 権威DNSサーバー運用者向け
    - NS・グルーレコードにおける短いTTLの危険性、
      権威DNSサーバーにおける攻撃の検知・対応について記述
- 応用対策編（未公開）
  - 本日紹介した2本のI-Dの内容なども参考にしつつ、
    作成・公開予定

2. DNS水責め攻撃の概要と対策

このパートの構成

- 攻撃の特徴
- 攻撃の目的
- 攻撃名称の由来
- 攻撃のシナリオ
- 攻撃成立の理由
- 取りうる攻撃対策
- この攻撃における注意点

攻撃の特徴（1/2）

- 2014年1～2月頃から世界的に観測され始めた、DNSサーバーに対するDDoS攻撃の手法
  - 攻撃は現在も続いている模様
- 第三者のオープンリゾルバーやホームルーターを、攻撃の踏み台として悪用している
- しかし、DNS応答を攻撃に使っておらず、いわゆるDNSリフレクター（DNS Amp）攻撃ではない
  - この攻撃では送信元IPアドレスの詐称は必須ではない

攻撃の特徴(2/2)

- 攻撃対象のランダムなサブドメインの問い合わせが、キャッシュ／権威DNSサーバーに大量に到達
  - (random).www.example.*TLD*のAレコード

ランダムな文字列(サブドメイン)

攻撃対象のドメイン名

- カミンスキー型攻撃手法の問い合わせと同一
  - そのため、「ランダムDNSクエリー攻撃」「ランダムサブドメイン攻撃」などとも呼ばれている
- しかし、カミンスキー型攻撃手法では検出されるはずの、問い合わせに対応する偽の応答が検出されない
  - そのため、当初は攻撃の目的が判然としなかった

攻撃の目的(と考えられていること)

- 攻撃対象ドメイン名の権威DNSサーバーが、真の攻撃対象であったと考えられている
  - 攻撃対象ドメイン名の権威DNSサーバーを過負荷にし、攻撃対象サイトをアクセス不能の状態に陥らせる
- 有力な状況証拠
  - 攻撃対象になった数百のドメイン名の多くが、中国・台湾・香港関係のECサイトやカジノサイトなどの「中華系の金が動くサイト」であったと報告されている

だとすると、国内ISPは攻撃者の真の攻撃目標ではなかった？

- 5月から7月にかけ、国内の複数のISPにおいて「DDoS攻撃によるDNSの障害」が相次いで発生
  - この攻撃の巻き添えを食ったと言われている
  - 巻き添えが起こる仕組みについては後述

なぜ「水責め」と呼ばれるのか？

- 2014年2月にこの攻撃を報告した米国Secure64 Softwareが、公式ブログで命名
  - Water Torture: A Slow Drip DNS DDoS Attack
    <https://blog.secure64.com/?p=377>
    - Water Torture = 水責め（水攻めではないので注意）
- dns-operations MLでの関係者の発言やSecure64 Softwareの技術者から得た回答によると、「水責め拷問」、特に「中国式水責め拷問」に由来しているとのこと

「中国式水責め拷問」

- Wikipedia英語版：Chinese water torture
  <http://en.wikipedia.org/wiki/Chinese_water_torture>

> “Chinese water torture is a process in which water is slowly dripped onto a person's forehead, allegedly driving the restrained victim insane.”

（椅子に拷問相手を縛り付け、額にゆっくりと水滴を滴下する）

- 今回の攻撃形態や、攻撃により各DNSサーバーが徐々に追い込まれていくさまが、<u>この拷問を彷彿（ほうふつ）とさせる</u>
  - 多数のBot（送信元IPアドレス）を用いた低頻度の攻撃

攻撃のシナリオ：登場人物

①攻撃者

オープンリゾルバーのリスト

②Botnet

③オープンリゾルバー

⑥攻撃対象ドメイン名の権威DNSサーバー

ns1

ns2

ISP A

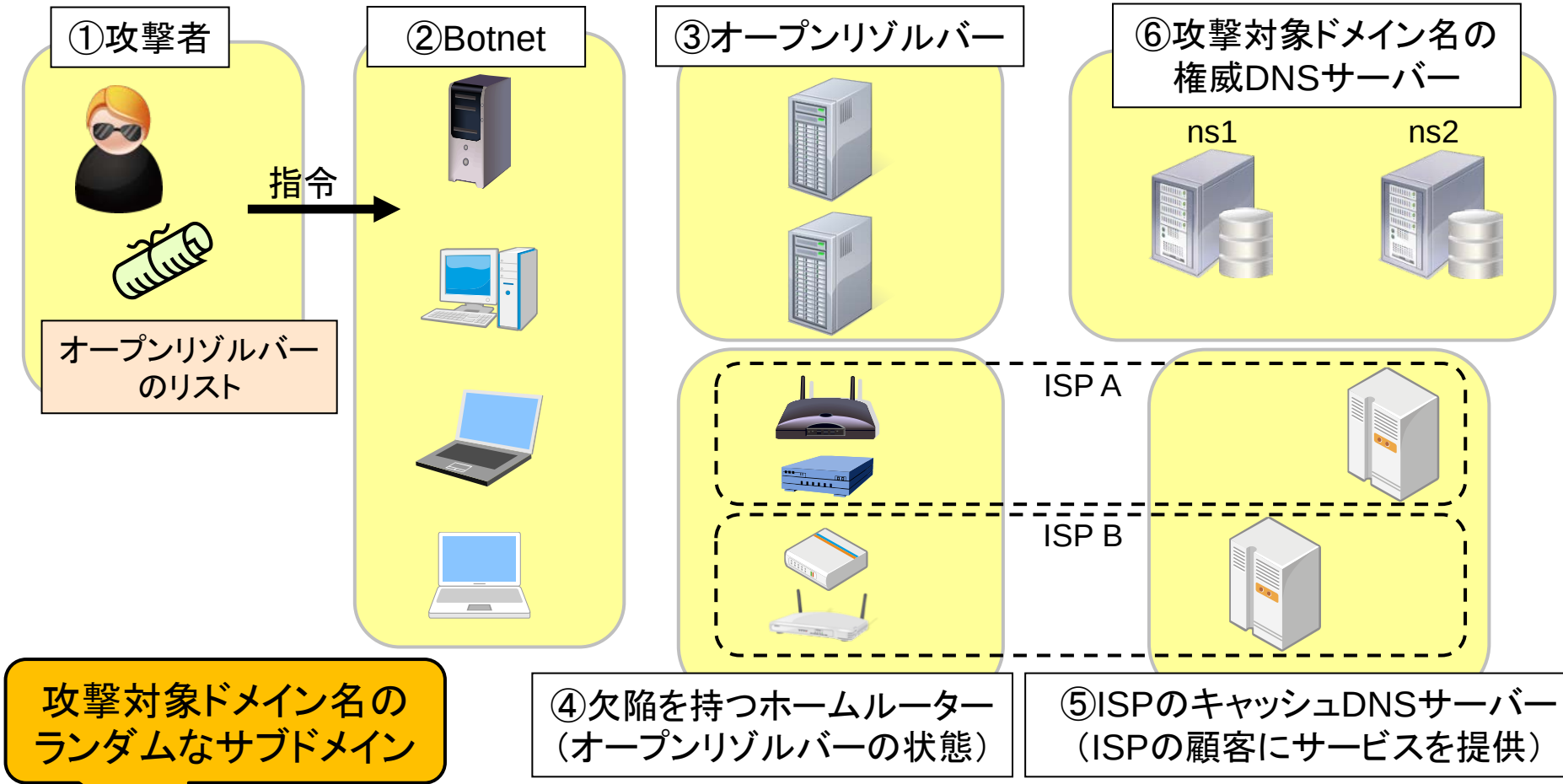
ISP B

④欠陥を持つホームルーター（オープンリゾルバーの状態）

⑤ISPのキャッシュDNSサーバー（ISPの顧客にサービスを提供）

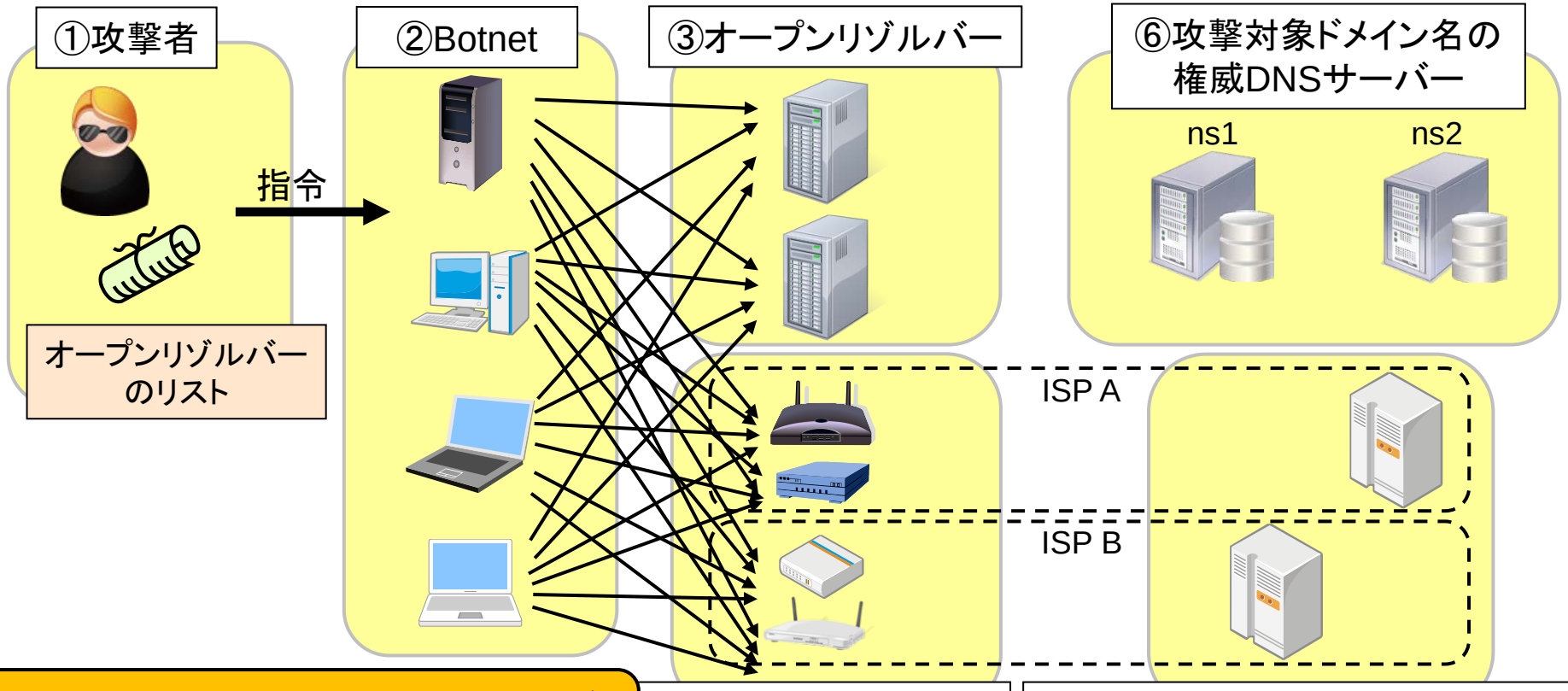
登場人物は大きく分けて6種類

攻撃のシナリオ（1/4）



1. 攻撃者がBotnetに、リストにあるオープンリゾルバーに対して
(random).www.example.*TLD*をDNS問い合わせするように指令を出す

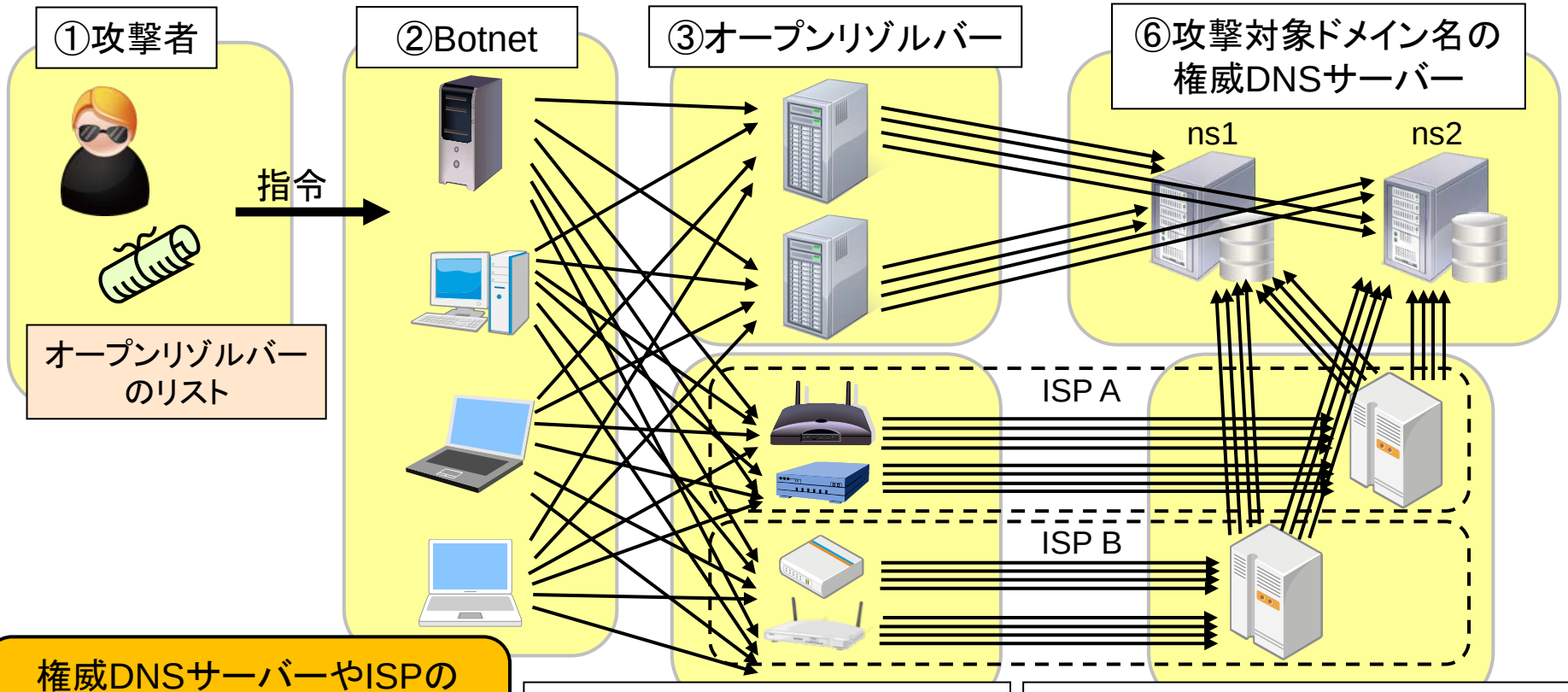
攻撃のシナリオ（2/4）



数十万台のBotから問い合わせが実施された例が報告されている

2. Botnetを構成する各PC（Bot）が、リストに掲載されたIPアドレスに問い合わせを送る
規制回避のため、数多くのBotから「広く薄く」問い合わせが送られる（Slow Drip）

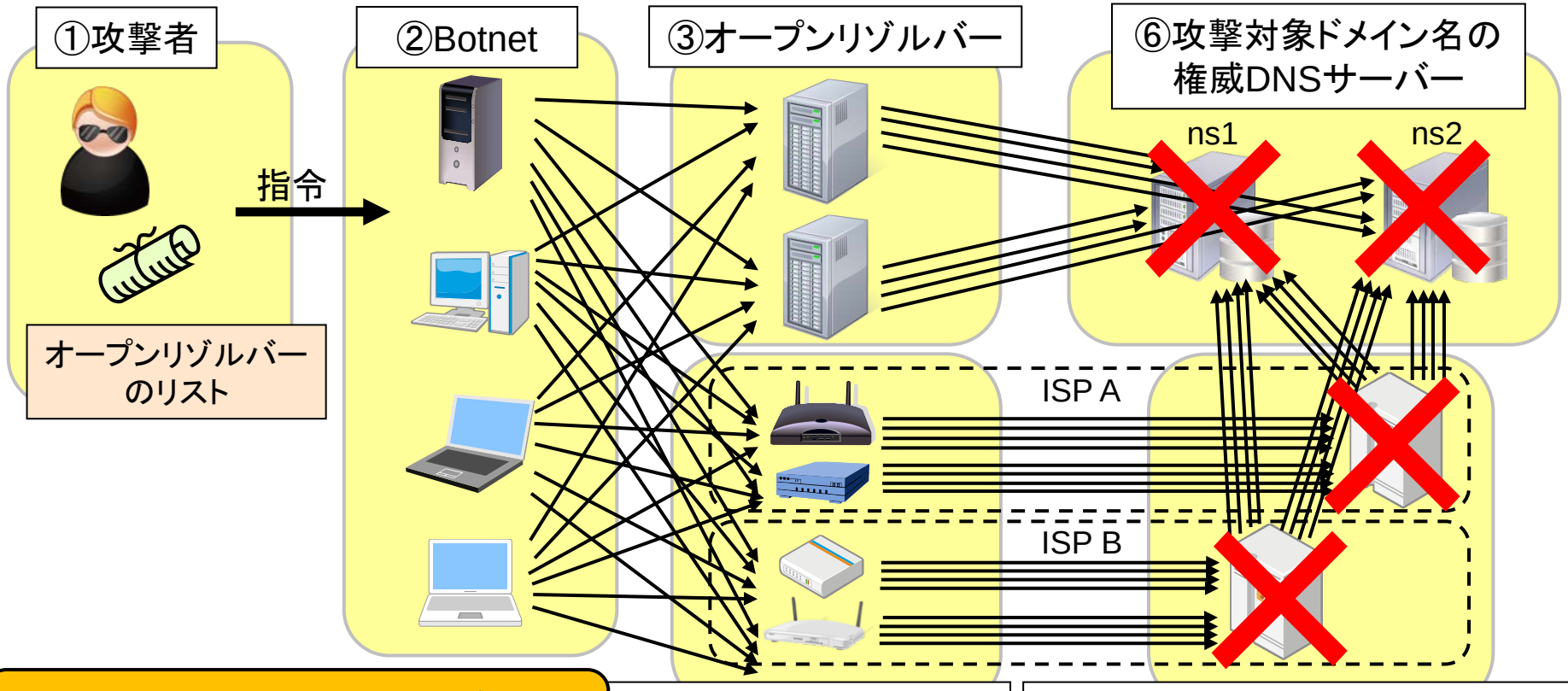
攻撃のシナリオ（3/4）



権威DNSサーバーやISPのキャッシュDNSサーバーにアクセスが集中する

3. キャッシュに存在しないため、権威DNSサーバーに問い合わせが毎回発生
（ホームルーターではISPのキャッシュDNSサーバーに問い合わせが毎回転送）

攻撃のシナリオ（4/4）



陥を持つホームルーター
ープンリゾルバーの状態）

⑤ISPのキャッシュDNSサーバー
（ISPの顧客にサービスを提供）

ISPのキャッシュDNSサーバーは、顧客側から攻撃されることになる

4. 問い合わせが集中する攻撃対象ドメイン名の権威DNSサーバーやISPのキャッシュDNSサーバーが過負荷になり、サービス不能状態に陥る

攻撃成立の理由

- 存在しない・キャッシュにない名前の処理はコスト高
  - 権威・キャッシュのいずれにとっても（特にキャッシュ）
  - 「キャッシュが効かないとどうなるか」を端的に表している
- 権威DNSサーバーの過負荷（応答遅延・無応答）が、キャッシュDNSサーバーにも悪影響を及ぼす
  - タイムアウトを待ったり、再送したり、別サーバーへの問い合わせを実施したりしなければならない
    - 処理中のセッションが溜まっていく
    - 限度を超えると、問い合わせを受け付けなくなる実装がある
- Botからの直接攻撃に比べ、フィルターしにくい
  - 正規のキャッシュDNSサーバーからのアクセスであるため

取りうる攻撃対策（1/3）キャッシュDNSサーバーにおける対策例

- 攻撃対象のゾーンをローカルに持たせる
  - 例：対象ドメイン名のIPアドレスとして127.0.0.1を指定
    - 対象ドメイン名に対するDoSは成立していることに注意
- BIND 9のRPZ（Response Policy Zone）機能を用い、「*.攻撃対象ドメイン名」の問い合わせに関する特別ルールを記述する
- iptables、あるいはそれに相当する機能でマッチングルールを書き、当該のDNS問い合わせを捨てる
  - 上記二つの対策も対象ドメイン名のDoSを成立させてしまう

決定打と考えられる対策はまだ存在しない

取りうる攻撃対策(2/3)
ISPの網側における対策例

事後資料で更新(IP123Bに関する補足を追加)

- IP53B(外部から53/udpへのアクセスをブロック)
  - ホームルーターの欠陥を外部から利用できなくする
  - リフレクター攻撃対策として、IP123B(NTP)と共に導入が図られつつある
    - 補足:IP123Bでは、123/udp⇔123/udpという通信が発生しうることに配慮する必要あり
- 「通信の秘密」との関係を考慮する必要あり
  - 詳細はこのあとのDNS DAYで

取りうる攻撃対策(3/3)
プログラムの追加導入・機能追加

- いくつかの対策用プログラムが発表されている
  - 「攻撃の検知」と「対応の自動化」を図るものが多い
    - dns_servfail_attack_mitigator
      - <https://github.com/cejennings/dns_servfail_attack_mitigator>
    - unbound-reqmon
      - <https://github.com/tarko/unbound-reqmon>
    - dnsbff
      - <https://github.com/willt/dnsbff>
- BIND 9.11でExperimentalな実装が入る見込み
  - 9月のUKNOFの発表で案のいくつかが発表された
    - Subscription版のBIND 9には一部実装されている
      - 今日のDNSOPS.JP BoFで発表されるかも？

この攻撃における注意点(1/2)

- この攻撃手法は単純かつ応用範囲が広い
  - DNSの仕組みそのものを攻撃に悪用している
  - キャッシュDNSサーバーへのDoSを目的にできうる
  - オープンリゾルバー経由やホームルーター経由でなくても成立しうる(クライアントPCのマルウェア経由など)
- 不用意な対策が悪影響を及ぼす場合がある
  - キャッシュDNSサーバーにおける対策により、攻撃対象ドメインへのDoSが成立してしまう

この攻撃における注意点(2/2)

- DNS RRL(Response Rate Limiting)の導入が、この攻撃による影響を大きくする場合がある
  - DNS RRLによる応答廃棄やTCPでの再送依頼が、キャッシュDNSサーバーの負荷を高める
- キャッシュポイズニング攻撃と併用が可能である
  - 現在の状況は「木を隠すなら森」の状態かもしれない

3. 未熟なDNSと
今後どう付き合うべきか

DNSに存在する様々な脅威（threat）

- 脅威（threat）：エラーやトラブルの直接の原因ではないが、その要因となりうるさまざまな要素
  - 参考：実生活における脅威の例
    - 焦っている、疲れている、時間に追われている、長時間の連続勤務、二日酔い、寝不足など
- 今日の話で出た以下の項目も、DNSの脅威の一つ
  - 委任の仕様、NSレコードの取り扱い、キャッシュの無力化
- DNSプロトコルに存在する脅威の例（RFC 3833より）
  - パケット傍受、ID・問い合わせの推測、名前の連鎖、不在証明、ワイルドカード、DNSSECの弱点

最近思うこと：DNSと ヒューマンファクターの高い共通性

※以降、水色部分はWikipediaより引用

- ヒューマンファクターとは

> 人間や組織・機械・設備等で構成されるシステムが、安全かつ経済的に動作・運用できるために考慮しなければならない人間側の要因のこと。

- ヒューマンファクターの重要な要素：スレットマネージメント（threat management：脅威の管理）
  - スレットマネージメントの3つのステップ

> - 見つける：適度な警戒心をもって監視を行い、スレットを発見したらその影響を予測する。
> - 避ける：発見したスレットに対してどのように対処するか、仲間・同僚間で認識を共有する。
> - とらわれない：突然発生・発見したスレットに対してはそのことにとらわれないで重要であるかどうかを見定める。

未熟なDNSと今後どう付き合うべきか

- 多くの脅威が存在するDNSには、ヒューマンファクターにおける管理手法の多くが適用できるような気がしている
  - ヒューマンファクターにおける「人間」を「DNS」に置き換えて考えると、しっくりくるものが多い（個人的な感覚）
- キーワード：「エラーと共存し、コントロールする」

どんな対策を講じても、どんなに教育や訓練を受けたとしてもヒューマンエラーを完全になくすことは不可能であるため、エラーと共存し、コントロールすることによって被害を最小限に留めることを主眼に置いている。

「共存し、コントロールする」ために必要なこと

- 脅威を減らすための努力の継続
  - 三大要素（仕様・実装・運用）全てにおいて必要
- 未熟なものであるという認識の共有
  - 組織内やオペレーター間、最終的には分野／レイヤーを越えた認識の共有が必要
- 分野／レイヤーを超えた協働・取り組み
  - いわゆる「専門家」や「ネットワーク屋」以外との協働・コーディネーションが必要

「それは未熟なものである」という認識の共有と、それを認識した上での広い範囲での協働・取り組みが重要

そして、これはきっとDNSだけに限らない

That’s it!