LinuC

LinuC レベル 2 Version10.0
技術解説セミナー

2025/11/29 開催

【例題解説】主題2.08：ドメインネームサーバー

INTERNOUS

インターノウス株式会社

（LPI-Japanアカデミック認定校）

竹本 季史

LPI-JAPAN

会社紹介、自己紹介

■会社紹介：インターノウス株式会社

- ●人材紹介サービス、人材派遣/SESサービス、IT未経験者の教育及び就職支援サービス、法人研修サービス
- ●未経験からインフラエンジニアやプログラマーになりたい方へ、無料で研修と就職支援サービスを行っています。
  プログラマカレッジ無料オンライン説明会

■自己紹介：竹本 季史(たけもと ときふみ)

- ●IT業界で約10年間勤務後、インターノウス株式会社エンジニアカレッジ講師。
- ●これまで約1000人以上を未経験者からエンジニアに養成。Linuxサーバー(メール、OpenSSH、シェルスクリプト、DB、監視、演習)を担当。

Linux技術者認定「LinuC（リナック)」とは

■LinuCとは

クラウド／DX時代のITエンジニアに求められるシステム構築から運用管理に必要なスキルを証明する技術者認定です。

✓クラウド活用に役立つスキルの習得

- オンプレミス／仮想化・コンテナを問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解

✓習得できるスキルが実践的

問題作成にはトップエンジニアも参加するコミュニティ内の意見を取り込むことで、本当に必要な内容を網羅的に盛り込んでいます。

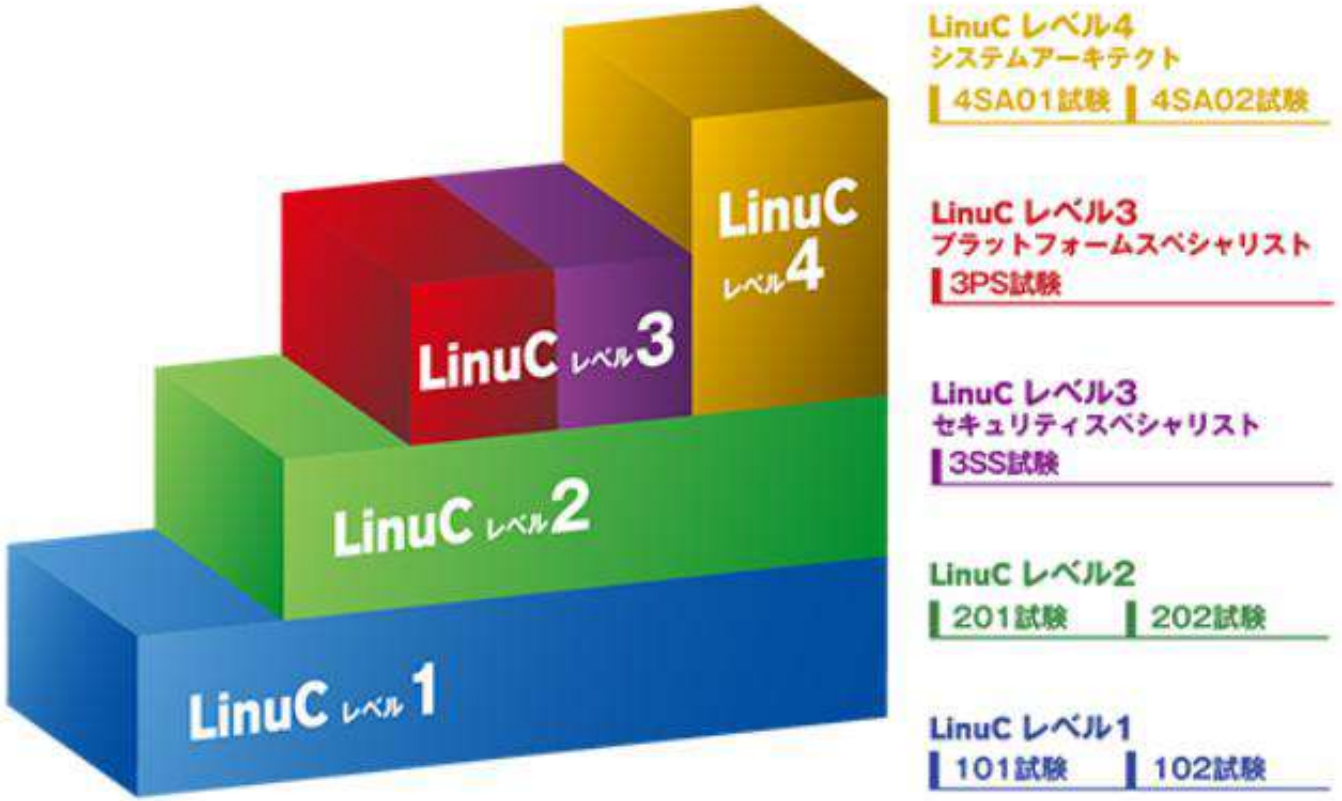
✓上流工程を担うアーキテクトの領域までカバー

システムの運用管理からアーキテクチャ設計までの４つのレベルをひとつずつ習得していくことで、活躍できるエンジニアとして必要なスキルを網羅的に身につけていくことができます

LinuC
Linux技術者認定

LinuCの４つのレベル

LinuCは、サーバーの運用管理からアーキテクト設計まで、システム開発・運用に必要な知識とスキルを体系立てて習得することができます。



本セミナーについて

- **解説する主題・内容**
  - LPI-Japanサイトの例題を取り上げて、LinuCレベル2 202試験 主題2.08のポイントを解説します
- **参加者の想定スキルレベル**
  - LinuCレベル2の受験を考えている方
- **セミナーのゴール**
  - 例題のポイントをつかんで、背景知識を強化する
  - 例題を解くことで、設定ファイルの中身とDNSの動作を結び付けて考えられる
  - デモを通じて設定ファイルの記述やコマンドの動作をつかむ

本セミナーについて

01

DNSの基礎知識

- DNSの基本用語
- ドメインの階層構造
- BIND, 主要ファイル、主要コマンド

02

2.08.1 BINDの設定と管理(例題2問)

named.confの構文とアクセス制御を学びます

- allow-recursion
- acl

03

2.08.2 ゾーン情報の管理(例題2問)

ゾーンファイルとリソースレコードを理解します

- SOA
- MX
- A など

04

2.08.3 セキュアなDNSサーバーの実現(例題1問)

DNSのセキュリティ対策を学びます

- DNSSEC

DNSの基本用語

名前解決とDNSの役割

人はIPアドレスを覚えにくい
そのため、DNSにホスト名+ドメイン名を登録しておき、
DNSに名前解決(ホスト名 ⇔ IPアドレス)をしてもらう

ドメインとは

DNSが名前を管理する範囲のこと。
ドメインの中には権威DNSサーバがある。
権威DNSのゾーン情報で、ドメイン内の
ホスト名、IPアドレスなどを管理する

2つのDNSサーバ

- 権威DNSサーバ(コンテンツサーバ)：ゾーン情報を**「持っている」**サーバ
- キャッシュDNSサーバ(フルリゾルバ)：ゾーン情報を**「探しに行く」**サーバ

DNSの基礎知識 ドメインの階層構造

internous.co.jp.ドメインの階層構造

ルート(.)

→ TLD(.jp)

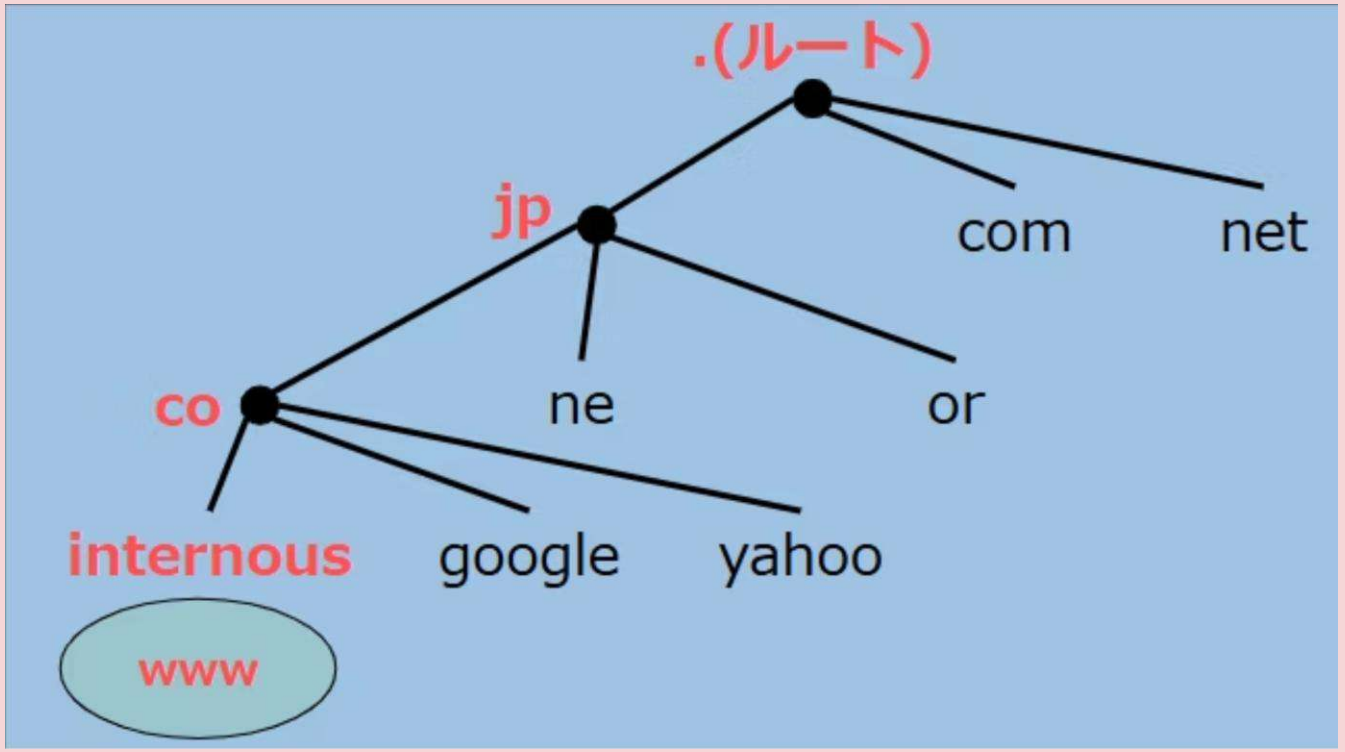
→ 第2レベル(.co)

→ 第3レベル(.internous)

・階層構造によりアクセス負荷や管理負荷を分散

・各ドメインに権威DNSサーバ

・自身のドメインのホスト名のみ管理



DNSの基礎知識 BIND, 主要ファイル, 主要コマンド

BIND (Berkeley Internet Name Domain)

- Linux環境で最も広く利用されるDNSサーバのソフトウェア。
- オープンソースで信頼性が高く、大規模な環境でも使われる。

主要ファイル

- **/etc/named.conf:** DNSサーバーの全体設定を定義するメイン構成ファイル。
  リスニングポート、ログ設定、ゾーン定義（管理するドメインの情報）、読み込むゾーンファイルの指定、アクセス制御などが含まれる。
- **ゾーンファイル:** 各ドメインに関する具体的なDNSレコード（A、MX、CNAME、NS、PTR、SOAなど）が記述されているファイル。
  named.confから参照され、ドメイン名とIPアドレスのマッピングやメールサーバー情報などを提供する。デフォルトで**/var/named**に存在する。

主要コマンド

- **named-checkconf**: /etc/named.confファイルの構文が正しいかを確認するためのコマンド。BINDの起動前にエラーを検出するのに役立つ。
- **named-checkzone:** ゾーンファイルの構文と整合性を検証するためのコマンド。
- **dig** (Domain Information Groper): DNSサーバーへの問い合わせを行う強力なコマンドラインツール。 DNSの名前解決の動作確認やトラブルシューティングに広く利用される。
- **rndc** (Remote Name Daemon Control): 実行中のBINDデーモンをリモートから制御するためのツール。 設定のリロード、キャッシュのダンプ、統計情報の表示など、BINDの運用管理に使用される。

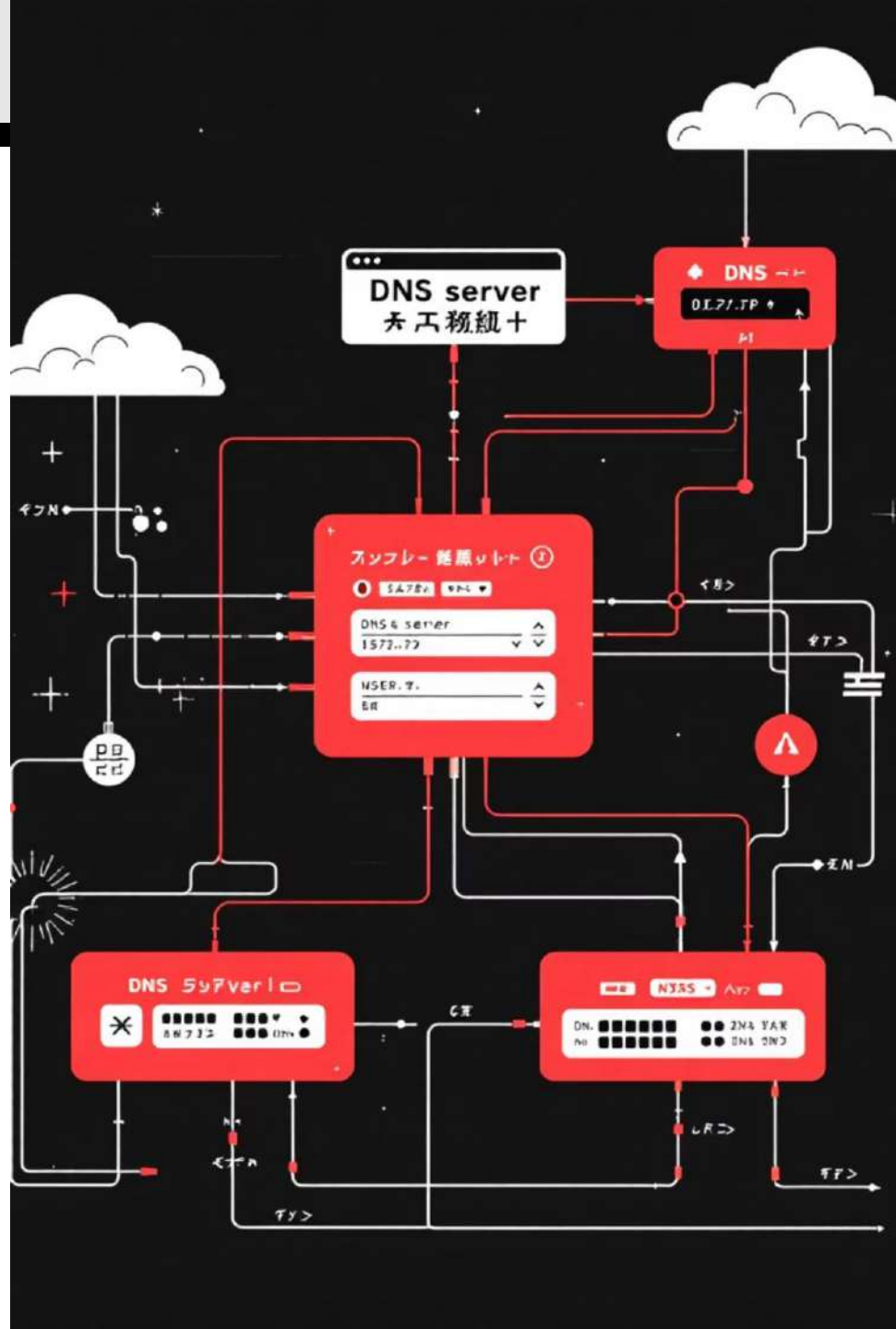
2.08.1 BINDの設定と管理

重要度: 3

学習範囲

- BIND の設定ファイル、用語、ユーティリティ
  - named.conf, host, dig, nslookup
- BIND の設定ファイルで、BINDゾーンファイルの位置を定義する。
  - named.conf
- 変更した設定ファイルおよびゾーンファイルの再読込
  - rndc, named-checkconf
- 代替ネームサーバーとしての dnsmasq, Unbound, NSD, PowerDNS について知っている。

- 【例題1】再帰問い合わせの制御
- 【例題2】named.confのステートメントの理解



【例題1】再帰問い合わせの制御

named.confで以下のような設定があります。

```
options {    __________ { 192.168.100.0/24; };
};
```

キャッシュDNSサーバとしてBINDが動作している場合、192.168.100.0/24からの再帰問い合わせのみを許可したいときに利用するオプションとして、下線部に当てはまるものは以下のうちどれでしょうか？

1. recursion
2. allow-recursion
3. allow-transfer
4. recursive-clients

問題の要点

用語の知識

- キャッシュDNSサーバー
- 再帰問い合わせ

設定項目の意味を考える

- 再帰問い合わせを許可(allow)するネットワークの指定
- 再帰 (recursion)の意味：ls -Rでlsをサブディレクトリがなくなるまで繰り返すイメージ。問い合わせるホスト名を管理する権威サーバの回答があるまで問い合わせを繰り返す。

【例題1 解説】再帰問い合わせの制御

正解: 2

```
allow-recursion { 192.168.1.0/24; };
```

解説

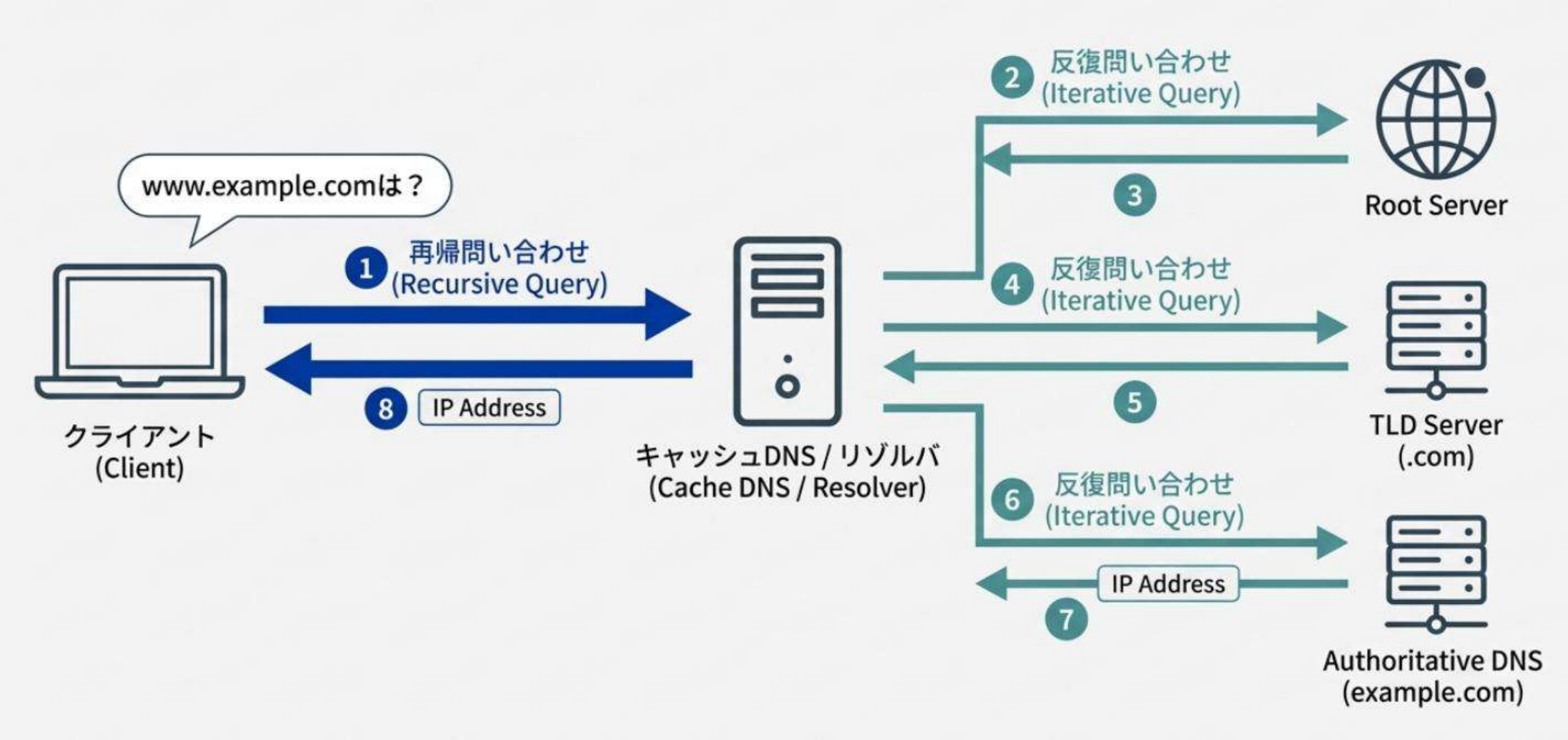
再帰問い合わせとは

再帰問い合わせは、クライアントのリクエストを受けたDNSサーバが最終の結果が得られるまで答えを探し回る。得られた回答はキャッシュされて次回以降の問い合わせ時に使われる（キャッシュDNSサーバ）反復問い合わせは権威DNS自身が管理するゾーンのみ回答する。

再帰問い合わせ先を許可する範囲

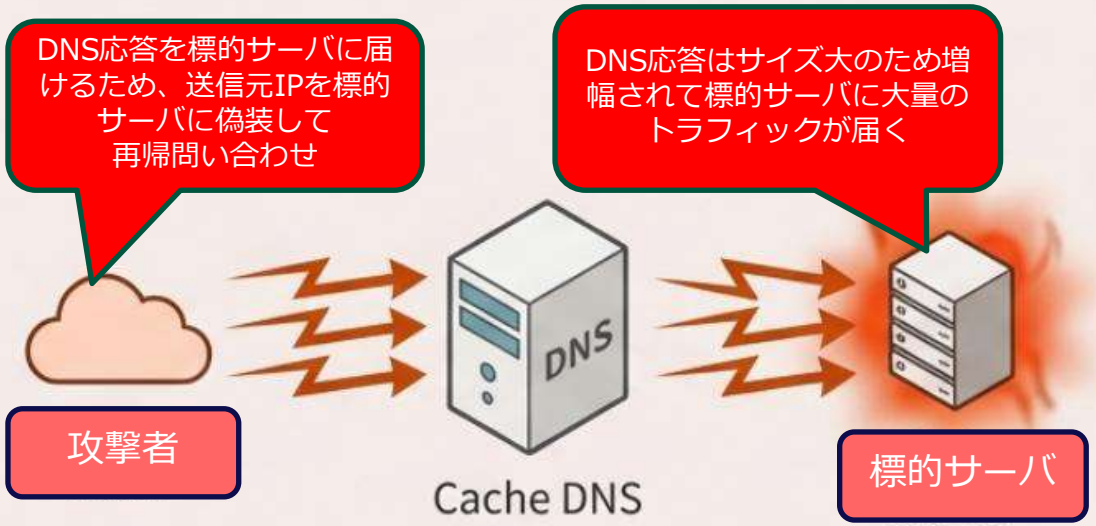
再帰問い合わせができる範囲を制限しないと、オープンリゾルバ(誰でも使える状態)となり、DNSアンプ攻撃の踏み台にされやすいため制限が必須。通常は社内LANのネットワークを指定。

再帰問い合わせの流れ


www.example.comは？
1 再帰問い合わせ
(Recursive Query)
8 IP Address
クライアント
(Client)
キャッシュDNS / リゾルバ
(Cache DNS / Resolver)
2 反復問い合わせ
(Iterative Query)
3
Root Server
4 反復問い合わせ
(Iterative Query)
5
TLD Server
(.com)
6 反復問い合わせ
(Iterative Query)
IP Address
7
Authoritative DNS
(example.com)

オープンリゾルバの危険性

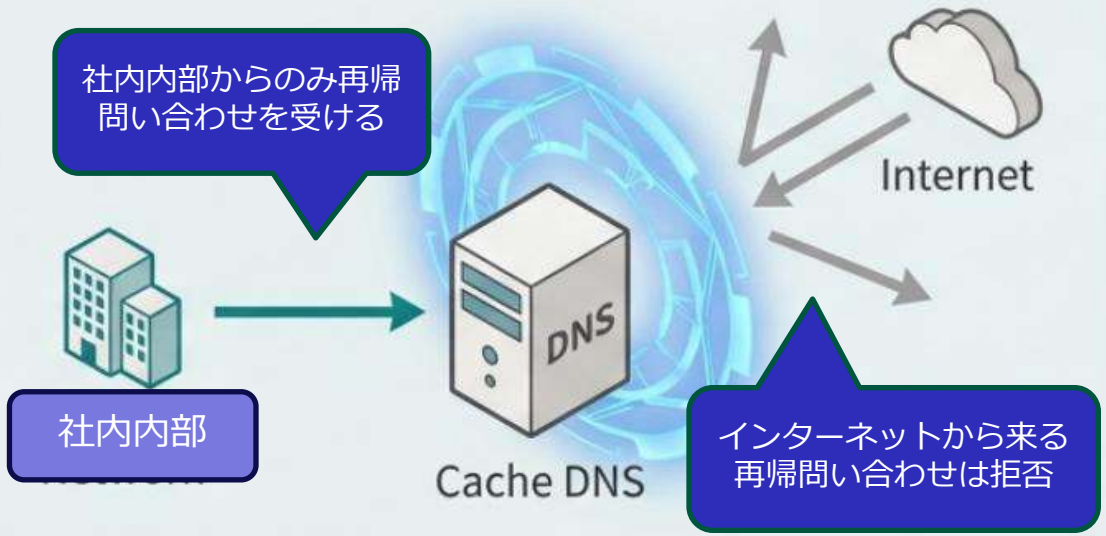
NG 危険 - オープンリゾルバ




```
allow-recursion { any; };
```

DNSアンプ攻撃の踏み台になってしまう

OK 安全 - 適切な設定




```
allow-recursion {
  localnets; };
```

安全な設定

デモ：allow-recursion範囲内、範囲外からの再帰問い合わせ

allow-recursionの**範囲内**から再帰問い合わせを実行して、 再帰問い合わせ**できる**ことを確認

allow-recursionの**範囲外**から再帰問い合わせを実行して、 再帰問い合わせ**できない**ことを確認

【例題2】named.confのステートメントの理解

named.confのステートメントと説明の組み合わせのうち、誤っているものを1つ選択せよ。

1. optionsステートメント：BINDのグローバルオプション設定を記述する。
2. aclステートメント：rndcコマンド(namedサービス管理)を使用するための設定を記述する。
3. zoneステートメント：管理対象のゾーンに関するゾーンファイルの場所やオプションなどを設定する。
4. includeステートメント：他の設定ファイルをnamed.confにインクルードできるようにする。

問題の要点

named.confの設定項目（ステートメント）の知識

- options
- acl
- zone
- include

ヒント

- 各ステートメントの意味を考える
- acl=Access Control Lists
- include = インクルード。含めるという意味

【例題2 解説】named.confのステートメントの理解

正解: 2

1

✖ **options**

BINDのグローバルオプション設定

2

⭕ **acl**

Access Control Lists(ACL)を定義する。
allow-query, allow-transfer, allow-recursionなどで使用可能。rndcを管理するのは**controls**

3

✖ **zone**

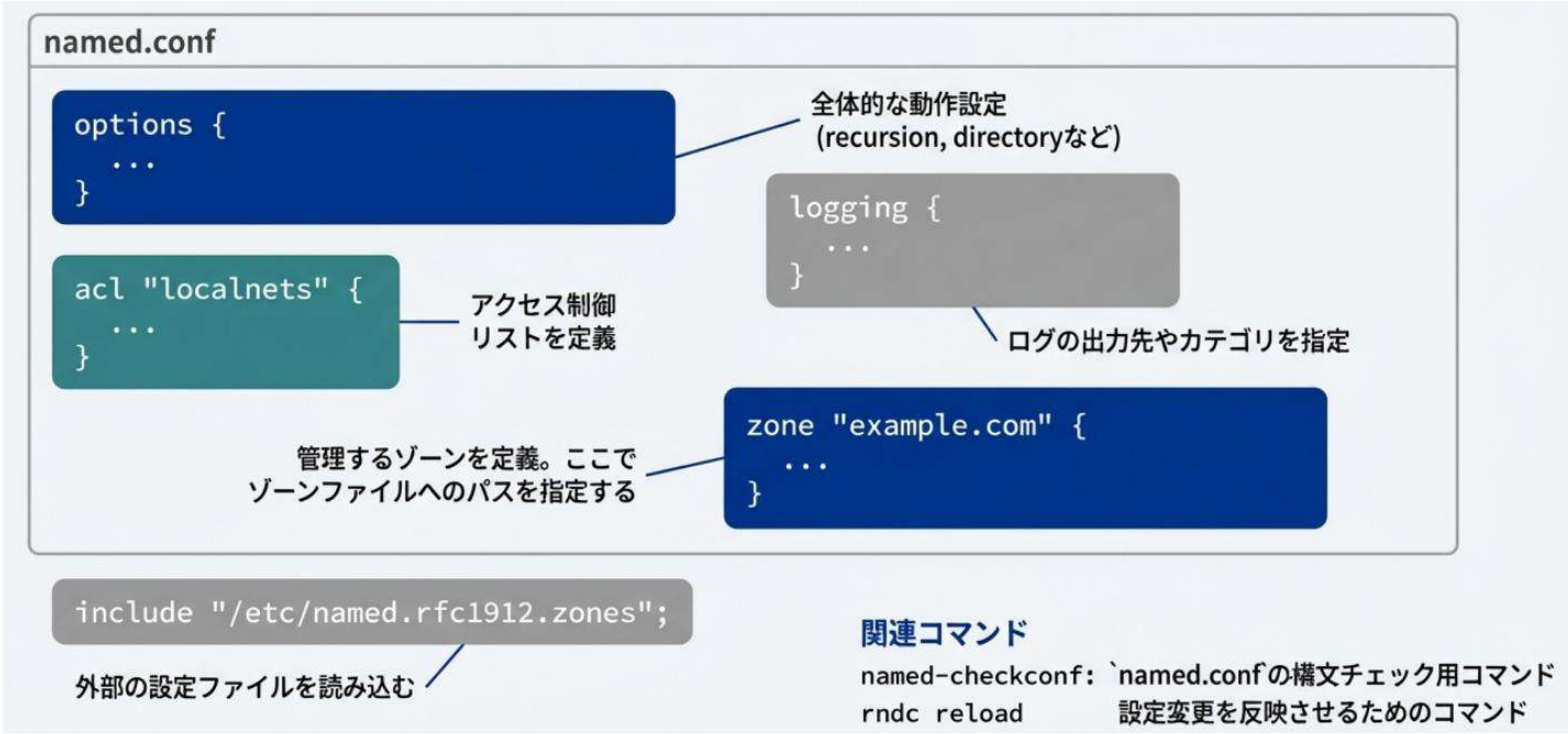
管理対象のゾーンファイルの場所やオプションなどを設定

4

✖ **include**

別の設定ファイルを読み込む (設定の分割管理)

【例題2 解説】named.confのステートメントの理解



デモ：aclの記述

```
//optionsステートメントの外に記述する
acl internal-net { 192.168.1.0/24; 10.0.0.0/8; 127.0.0.1; };
```

```
// allow-query, allow-transfer, allow-recursion などでaclで定義した名前を記述する
allow-query { internal-net; };
```

2.08.2 ゾーン情報の管理

重要度: 2

学習範囲

- BINDゾーンファイルのレイアウト、内容、ファイル配置
  - ゾーンファイルの書式, リソースレコードの書式
- 逆引きゾーンを含む、ゾーンファイルに新しいホストを追加する際の確認方法
  - named-compilezone, named-checkzone

- 【例題3】ゾーンファイルの基本構造
- 【例題4】ゾーンファイルのレコードの読み方

【例題3】ゾーンファイルの基本構造

以下の正引きゾーンファイルについて、正しいものを選択せよ。

```
$ORIGIN opensourcetech.test.
$TTL    604800
@       IN      SOA     dns.opensourcetech.test. root.opensourcetech.test. (
             2022051501     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
        IN      NS      dns.opensourcetech.test.
        IN      MX 10   mail.opensourcetech.test.
dns     IN      A       192.168.1.247
www     IN      A       192.168.1.247
mail    IN      A       192.168.1.247
ftp     IN      A       192.168.1.247
smb     IN      A       192.168.1.247
```

1. Aレコードに同じIPv4アドレスが割り当てているのは誤りである
2. "dns.opensourcetech.test."の最後の"."は省略可能である
3. MXレコードにある"10"は、メールサーバの優先度を表す
4. "root.opensourcetech.test."はDNSサーバの別名を表す

【例題3】問題の要点:各レコードの意味と末尾ドットの意味

主要リソースレコード（RR）

A

ホスト名 -> IPv4アドレス

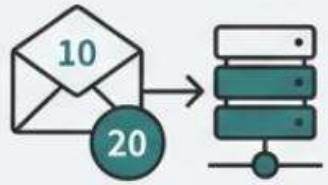
AAAA

ホスト名 -> IPv6アドレス

NS

ゾーンの権威DNSサーバを指定（委任）

MX



メール配送先のサーバを指定（数字が小さい方を優先）

CNAME

ホスト名の別名を定義

「.」の有無が運命を分ける

末尾ドットなし（相対ドメイン名）

```
www IN A 192.168.1.10
```

$ORIGIN (example.com.)

www.example.com.

自動的に $ORIGIN が末尾に付加される。

試験のひっかけ常連！
末尾ドットの有無で意味が全く異なることを理解する。

末尾ドットあり（FQDN：完全修飾ドメイン名）

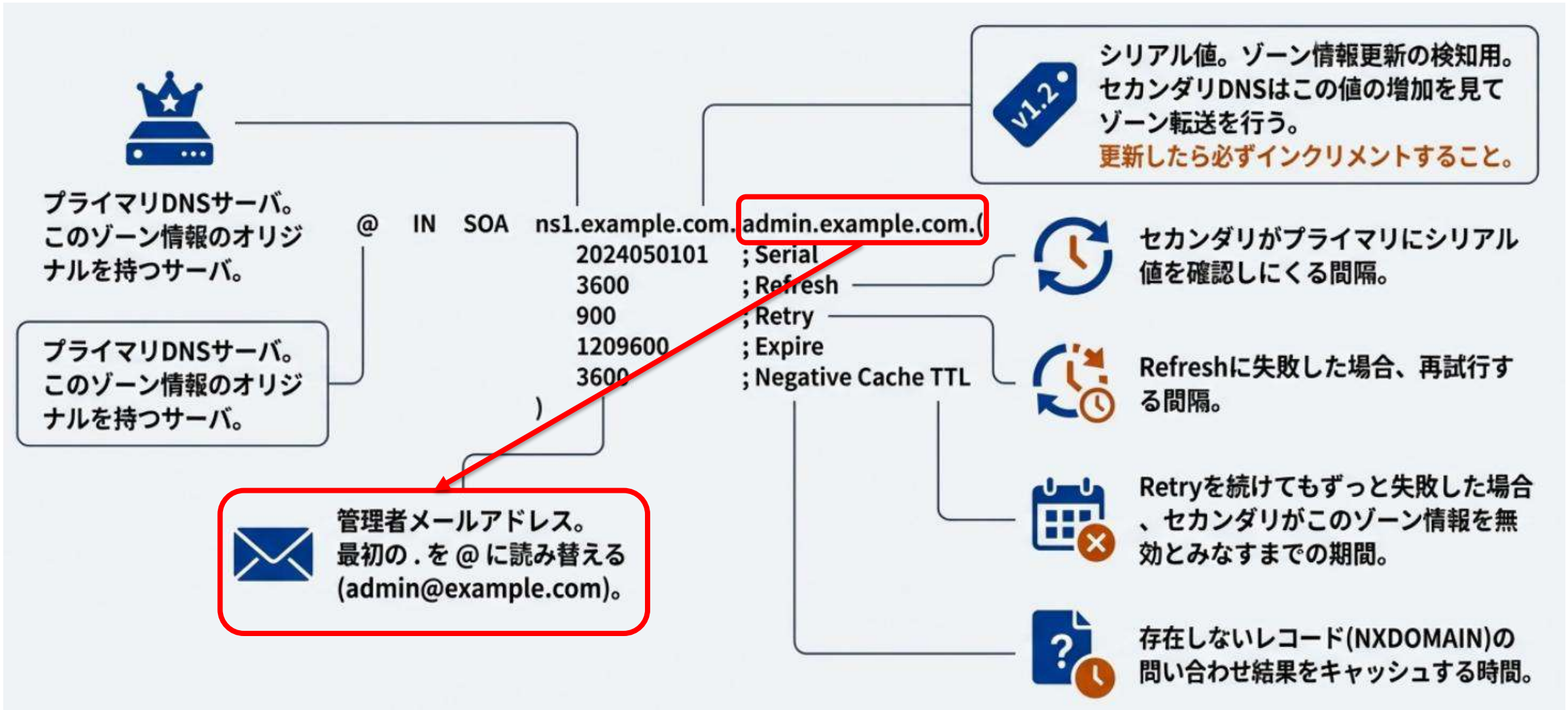
```
www.example.com. IN A 192.168.1.10
```

追加されない

www.example.com.

$ORIGIN は付加されず、記述したままの完全修飾ドメイン名となる。

【例題3】問題の要点:SOAレコードの意味



【例題3 解説】ゾーンファイルの基本構造

正解: 3. MXレコードにある"10"は、メールサーバの優先度を表す

1

✖ **Aレコードに同じIPv4アドレスが割り当てているのは誤りである**

ひとつのIPアドレスで用途ごとにホスト名を変えることができる。

- mail.example.com → メール用
- www.example.com → Web用
- dns.example.com → DNS用

小規模な企業で1台のサーバーで多数の役割を与えたい場合などに使用。

2

✖ **"dns.opensourcetech.test."の最後の"."は省略可能である**

ドメインの最後の「.」は「これ以上ドメインを補完しない、完全修飾名（FQDN）」という意味。
省略してしまうと、$ORIGINのドメインを補完してしまう。

3

⭕ **MXレコードにある"10"は、メールサーバの優先度を表す**

MXはドメインのメールサーバーのホスト名を指定するレコード。数字はメールサーバーの優先度を表す。MXと10の間はスペースが必要。

4

✖ **"root.opensourcetech.test."はDNSサーバの別名を表す**

該当の箇所は、SOAレコードで**ドメインの管理者のメールアドレス**を記述する。
本来ならroot@opensourcetech.test.となるはずだが、@自体が「このドメイン」を示すのでドットに置き換えてある。

【例題4】ゾーンファイルのレコードの読み方

mylinuc2.com のゾーンファイル(mylinuc2.com.zone)に以下のような記述があります。 この説明で正しいものを2つ指定してください。

```
1行目： @  IN SOA mylinuc2.com. root.mylinuc2.com.
(...省略...)
2行目：        NS ns.mylinuc2.com.
3行目：        MX 10 mail.mylinuc2.com.
4行目： ns     A 192.168.10.1
5行目： mail    A 192.168.10.2
6行目： host01  A 192.168.10.3
7行目： www    CNAME host01
```

1. このゾーンのネームサーバはns.mylinuc2.com.である
2. このゾーンのメールサーバはns.mylinuc2.com.である
3. ns.mylinuc2.com.のIPアドレスは192.168.10.2である
4. www.mylinuc2.com.はhost01.mylinuc2.com.の別名である

【例題4】(再掲)問題の要点:各レコードの意味

【例題4 解説】ゾーンファイルのレコードの読み方

正解:1. このゾーンのネームサーバはns.mylinuc2.com.である

4. www.mylinuc2.com.はhost01.mylinuc2.com.の別名である

1

○ このゾーンのネームサーバは ns.mylinuc2.com.である

ネームサーバはNSレコードを見る

2

✖ このゾーンのメールサーバは ns.mylinuc2.com.である

メールサーバはMXレコードを見る

3

✖ ns.mylinuc2.com.のIPアドレスは 192.168.10.2である

IPアドレスはホスト名のAレコードを見る

4

○ www.mylinuc2.com.はhost01.mylinuc2.com.の別名である

別名はCNAMEレコードで記述する。wwwのCNAMEレコードはhost01なので正解。

デモ：ゾーンファイルの記述

```
$ORIGIN engineer-college.jp.
$TTL    604800
@       IN      SOA     dns.engineer-college.jp. root.engineer-college.jp. (
                         2025112901     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
        IN      NS      dns.engineer-college.jp.
        IN      MX 10   mail.engineer-college.jp.
dns     IN      A       192.168.1.247
host01  IN      A       192.168.1.248
mail    IN      A       192.168.1.249
ftp     IN      A       192.168.1.247
smb     IN      A       192.168.1.247
www     IN      CNAME   host01
```

2.08.3 セキュアなDNSサーバーの実現

重要度: 2

学習範囲

- chroot 環境で稼働するようBINDを設定する。
- forwarders文を使用してBINDの構成を分割する。
  - named.conf
- DNSSEC および基本的なツールについて知っている。
  - dnssec-keygen, dnssec-signzone, TSIG(Transaction Signature)
- DANE および関連レコードについて知っている。

- 【例題5】DNSの改ざん検知

【例題5】DNSの改ざん検知

以下のうち、電子署名の仕組みを用いて、DNSクライアントがDNSサーバから送られてくるDNS情報が改竄されていないことを検証するものはどれか。

1. chroot jail
2. TSIG
3. DNSSEC
4. DANE

【例題5】DNSの改ざん検知

どの範囲のセキュリティなのかを理解する

- “電子署名（＝公開鍵暗号方式）を使って、DNS情報の改ざんを検知する仕組みはどれか？”
- いずれの選択肢もセキュリティの技術だが目的や守る範囲が違う。

略語の元になった言葉を知る

- chroot jail
  - chroot=change root ルートディレクトリを変える
  - jail=檻
- TSIG（**T**ransaction **SIG**nature）
  - トランザクション署名
- DNSSEC（**DNS Sec**urity Extensions）
  - DNSセキュリティ拡張
- DANE（**D**NS-based **A**uthentication of **N**amed **E**ntities）
  - 名前を持つエンティティ（サーバ証明書）をDNSで認証

【例題5 解説】ゾーンファイルの基本構造

正解: 3. DNSSEC

1

✖ **chroot jail**

namedプロセスを檻(/var/named/chroot)に入れて本物のルートディレクトリを見えなくする仕組み。
たとえ、namedの権限を奪取されても被害を最小限に抑えることができる。

2

✖ **TSIG**

マスタ・スレーブ構成のDNSサーバ間におけるゾーン転送をセキュアに行う仕組み。

3

⭕ **DNSSEC**

電子署名の仕組みを利用して、ゾーン情報が改ざんされていないことの検証と権威サーバの正当性を保証する。
これにより、キャッシュポイズニングを防ぐことができる。

4

✖ **DANE**

サーバ証明書の正当性を DNSSECを使って検証する仕組み。
CA の代わりに DNS を“信頼の根っこ”にする技術。

キャッシュポイズニングとは

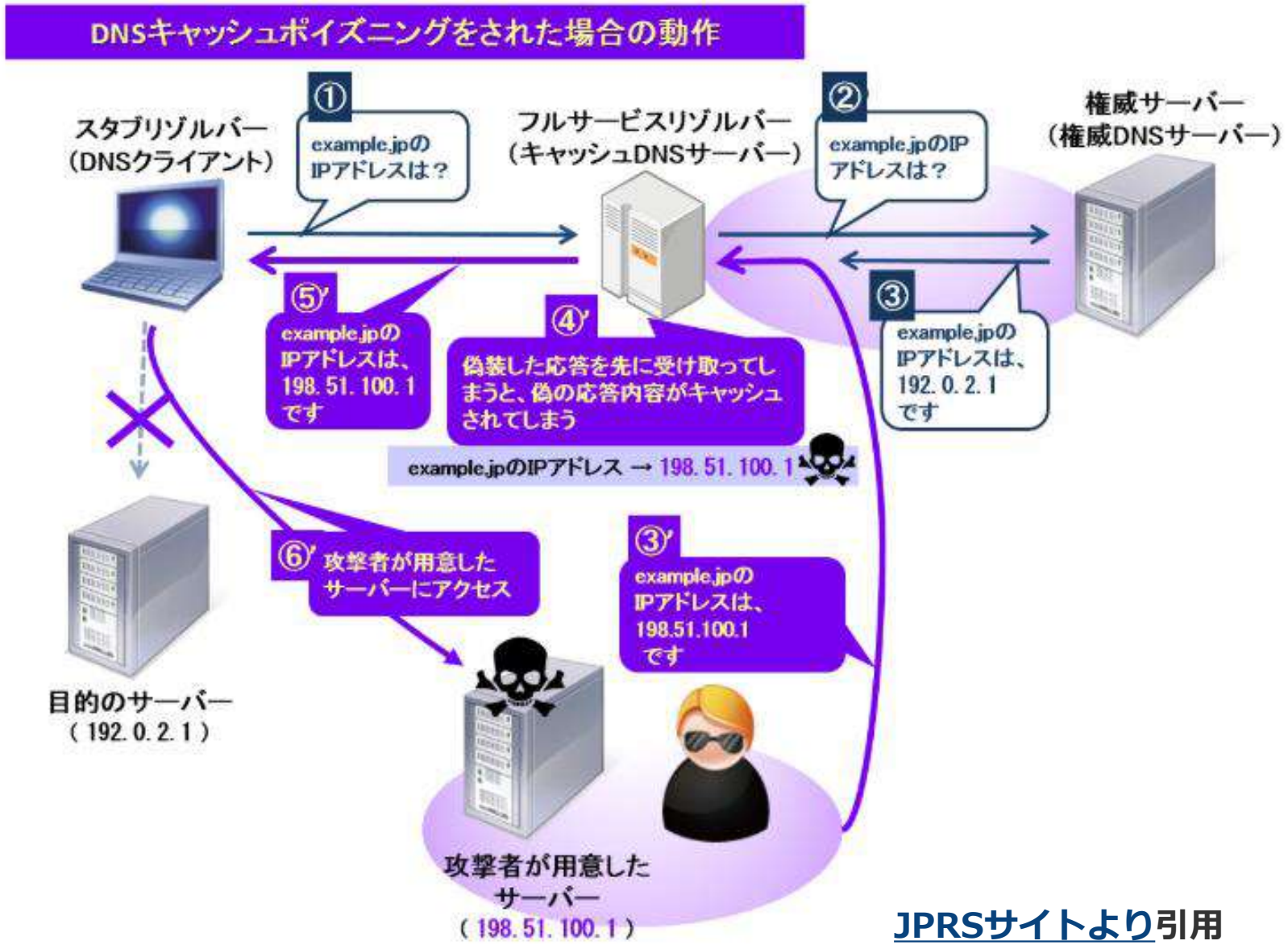
正常な流れは、DNSクライアントが再帰問い合わせをキャッシュDNSサーバに行い（①）、キャッシュDNSサーバは権威サーバに問い合わせて(②)、権威サーバからの応答を受け取ります(③)。

攻撃者は、権威サーバーからの応答（③）がキャッシュDNSサーバに届く前に発信元などを偽装した応答（③′）を送り込みます。

送り込みに成功した場合、
キャッシュDNSサーバに偽の応答内容がキャッシュされます（④′）。

DNSクライアントは、キャッシュされた偽の応答（⑤′）を受け取るため、攻撃者が用意した別のサーバーに誘導されます（⑥′）

以降、ユーザーのアクセスはキャッシュされた内容が無効になるまで攻撃者が用意したサーバーに誘導されることになります。



JPRSサイトより引用

DNSSECの目的

DNSSECの目的: DNSキャッシュポイズニングなどの攻撃を防ぎ、DNS応答の信頼性を確保する。

守るもの (Protects)

応答の完全性（改ざん検知）
ゾーン情報が途中で改ざんされていないことを保証する。

応答の出自認証（正当性）
応答を返した権威DNSサーバが正当な管理者であることを保証する。

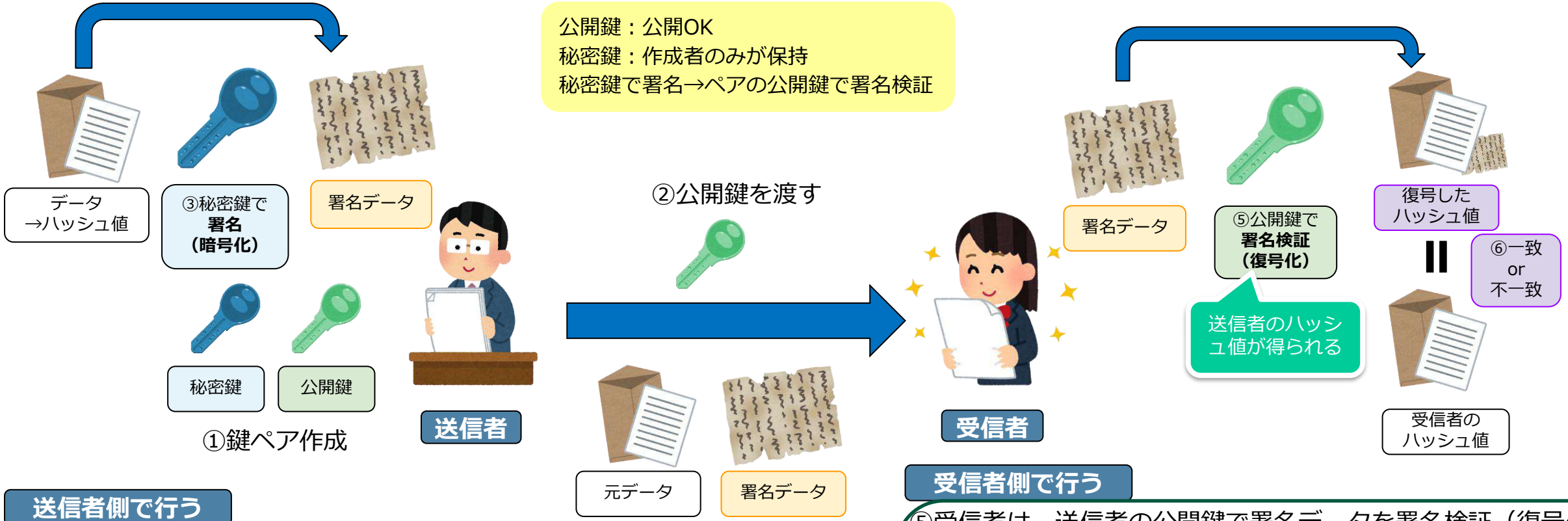
守らないもの (Does NOT Protect)

通信の暗号化
これはDoT（DNS over TLS）/ DoH（DNS over HTTPS）の役割。

プライバシー保護
問い合わせ内容自体は暗号化されない。

DNSSECの前提となる電子署名の仕組み

公開鍵：公開OK
秘密鍵：作成者のみが保持
秘密鍵で署名→ペアの公開鍵で署名検証



送信者側で行う

①送信者は、秘密鍵と公開鍵の鍵ペアを作成
②送信者は、受信者に公開鍵を渡す
③送信者は、データをハッシュ化した「ハッシュ値」を秘密鍵で署名（暗号化）して「署名データ」を作成
④送信者は、元データと署名データを渡す

受信者側で行う

⑤受信者は、送信者の公開鍵で署名データを署名検証（復号化）して「復号したハッシュ値」を得る
＝ペアの秘密鍵を持つ送信者によってデータが作成および送信されたこと（**真正性**）が証明される

⑥受信者は、元データをハッシュ化した「受信者のハッシュ値」と「復号したハッシュ値」が一致することを確認
＝途中で改ざんされていないこと（**完全性**）が証明される

DNSSEC仕組み：ZSK/KSK, 信頼の連鎖について

🔑 ZSK（Zone Signing Key）／KSK（Key Signing Key）

- **ZSK：ゾーン署名鍵**
- A・NS・MX など「ゾーン情報」を署名する
- ゾーン情報が改ざんされていないことを証明する

- **KSK：鍵署名鍵**
- ZSK/KSK公開鍵セット（DNSKEY）を署名する
- DNSSEC の「公開鍵を正しいと証明する鍵」
- 親ゾーンにKSKのハッシュ値（=*DS*）を登録して信頼の連鎖をつくる

🔗 信頼の連鎖（Chain of Trust）

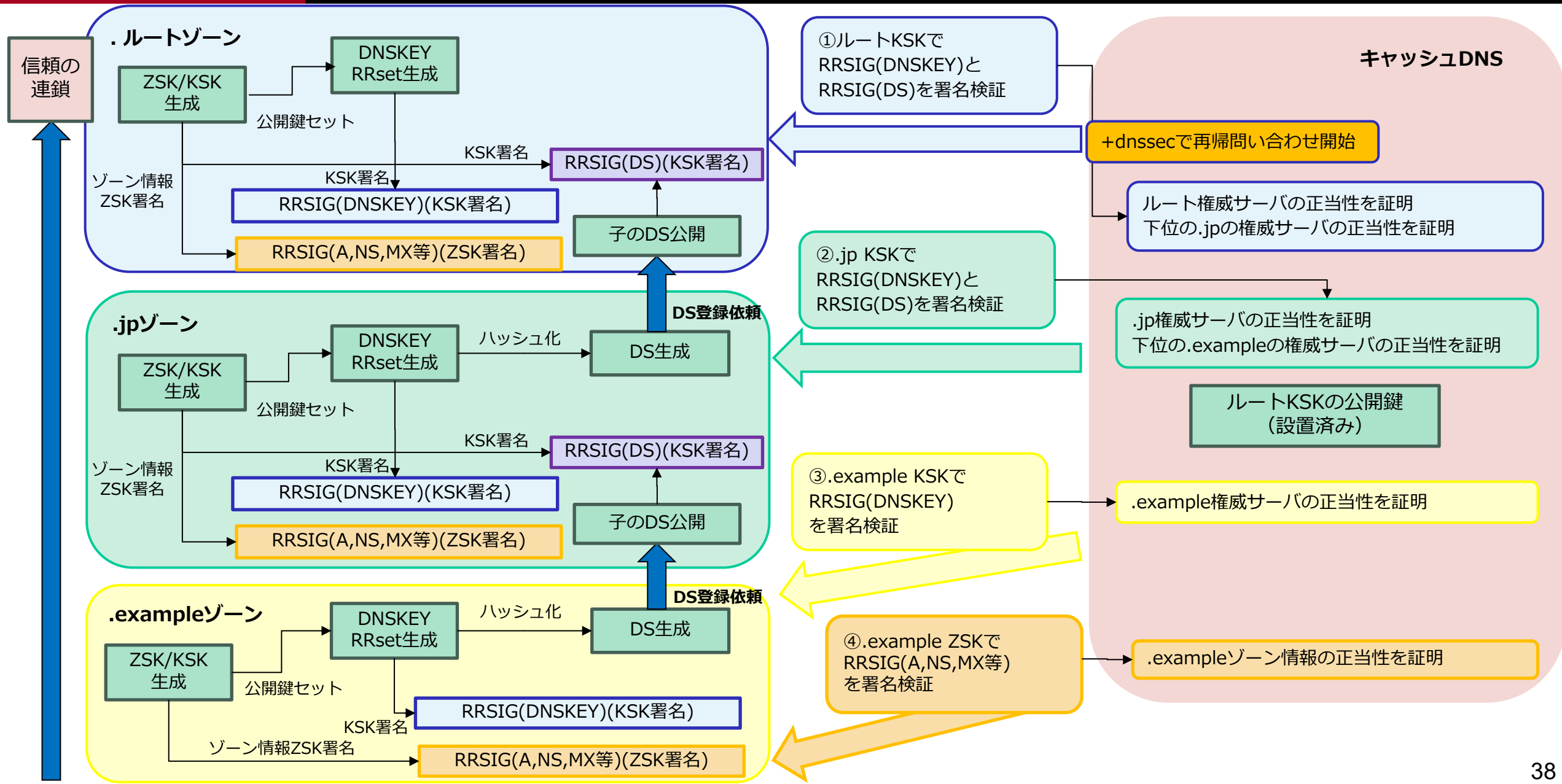
DNSSEC 全体の根っこになる概念。

KSKとZSKを使用して、

DNSキャッシュサーバ
→ルート
→ .jp
→ example.jp
→ example.jpのゾーン情報

と上位の権威サーバが下位ゾーンの権威サーバを信頼していき、
最終的に問い合わせ先のゾーン情報まで、信頼が連鎖する概念。

DNSSECの仕組み


信頼の連鎖
. ルートゾーン
ZSK/KSK生成
DNSKEY RRset生成
公開鍵セット
KSK署名
KSK署名
ゾーン情報ZSK署名
RRSIG(DS)(KSK署名)
RRSIG(DNSKEY)(KSK署名)
RRSIG(A,NS,MX等)(ZSK署名)
子のDS公開
DS登録依頼
.jpゾーン
ZSK/KSK生成
DNSKEY RRset生成
ハッシュ化
DS生成
公開鍵セット
KSK署名
KSK署名
ゾーン情報ZSK署名
RRSIG(DS)(KSK署名)
RRSIG(DNSKEY)(KSK署名)
RRSIG(A,NS,MX等)(ZSK署名)
子のDS公開
DS登録依頼
.exampleゾーン
ZSK/KSK生成
DNSKEY RRset生成
ハッシュ化
DS生成
公開鍵セット
KSK署名
ゾーン情報ZSK署名
RRSIG(DNSKEY)(KSK署名)
RRSIG(A,NS,MX等)(ZSK署名)
①ルートKSKで RRSIG(DNSKEY)と RRSIG(DS)を署名検証
キャッシュDNS
+dnssecで再帰問い合わせ開始
ルート権威サーバの正当性を証明
下位の.jpの権威サーバの正当性を証明
②.jp KSKで RRSIG(DNSKEY)と RRSIG(DS)を署名検証
.jp権威サーバの正当性を証明
下位の.exampleの権威サーバの正当性を証明
ルートKSKの公開鍵（設置済み）
③.example KSKで RRSIG(DNSKEY) を署名検証
.example権威サーバの正当性を証明
④.example ZSKで RRSIG(A,NS,MX等) を署名検証
.exampleゾーン情報の正当性を証明

デモ：DNSSECで使用するコマンド

鍵ペアを生成するdnssec-keygenと署名を行うdnssec-signzoneコマンドが試験範囲となっています。

```
#ZSKとKSKの鍵ペア生成
dnssec-keygen -a RSASHA256 -b 2048 -n ZONE engineer-college.jp
dnssec-keygen -a RSASHA256 -b 4096 -n ZONE -f KSK engineer-college.jp

#ゾーンファイルへZSKとKSKの公開鍵DNSKEYを追加
cat Kengineer-college.jp.*.key >> engineer-college.jp.zone
cat engineer-college.jp.zone

#ZSKの秘密鍵でRRsetを署名、KSKの秘密鍵でDNSKEY RRsetを署名
dnssec-signzone -o engineer-college.jp engineer-college.jp.zone

#各RRsetが署名されて、RRSIGになっている
cat engineer-college.jp.zone.signed

#DSレコード(DNSKEYのハッシュ値)が見える
cat dsset-engineer-college.jp.
```

まとめ

- **セミナーのゴール（再掲）**
  - 例題のポイントをつかんで、背景知識を強化する
  - 例題を解くことで、設定ファイルの中身とDNSの動作を結び付けて考えられる
  - デモを通じて設定ファイルの記述やコマンドの動作をつかむ

ご清聴ありがとうございました。