

## DNS キャッシュポイズニング対策 ～DNSの役割と関連ツールの使い方～

1. DNSキャッシュポイズニング
2. DNSの動作と関連ツール
3. 検査ツールの使い方と注意点
4. 再帰動作の設定

2009年8月

独立行政法人情報処理推進機構セキュリティセンター



INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

# 1. DNSキャッシュポイズニング

## 1.1 DNSの仕組み

## 1.2 DNSキャッシュポイズニング

DNS(Domain Name System)とは、

DNSは、ホスト名(例:www.ipa.go.jp)とIPアドレス(例:202.229.63.242)とを紐付ける情報を提供します。Webサーバへのアクセスやメール送受信など、インターネット上の多くのアプリケーションはDNSを前提としていることから、DNSはインターネットの基盤サービスとも言われています。

DNSキャッシュポイズニングとは、

DNSキャッシュポイズニングは、DNSサービスを提供しているサーバ(DNSサーバ)に偽の情報を覚えこませる攻撃手法です。攻撃が成功すると、DNSサーバは覚えた偽の情報を提供してしまうことになります。このため、ユーザは正しいホスト名(例:www.ipa.go.jp)のWebサーバに接続しているつもりでも、提供された偽の情報により、攻撃者が罠をはったWebサーバに誘導されてしまうことになります。

このような危険性から身を守るためにも、DNSとDNSキャッシュポイズニングの特性を理解し、安全なDNSサーバによる基盤サービスを実現しましょう。

## DNSの役割

DNSサーバは、ホスト名(例: www.ipa.go.jp)をIPアドレス(例: 202.229.63.242)に変換したり、ドメイン(例: ipa.go.jp)で利用するメールサーバ(例: ipa-ns.ipa.go.jp)を教えたりするなどの役割を担っています。

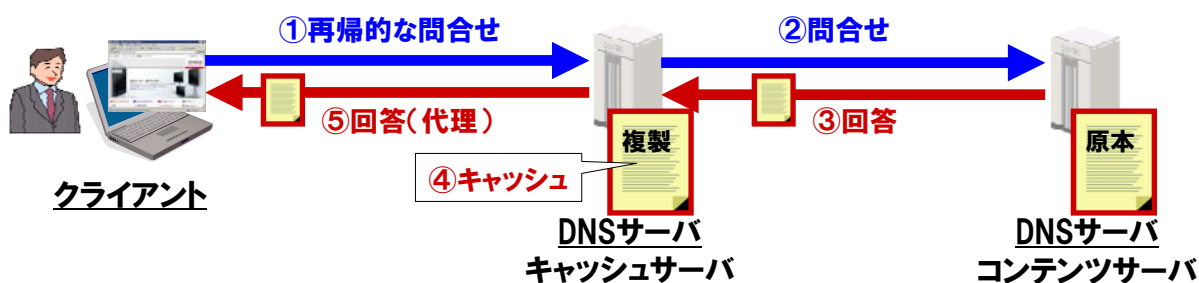


クライアントでは、手動あるいは、自動(DHCPなど)で設定されたDNSサーバに問合せを行い、ホスト名をIPアドレスに変換したり、ドメインで利用するメールサーバなどの情報を得たりしています。このような操作のことをDNSサーバによる名前解決と呼びます。ここで、ホストとはサーバやクライアントなどコンピュータを総称で、ホスト名はサーバやクライアントなどコンピュータに付けられた名前のことです。

## DNSサービスを実現するサーバ機能

DNSサーバには、「コンテンツサーバ」と「キャッシュサーバ」の2種類があり、これらが連携してDNSサービスを実現しています。

- ◇ コンテンツサーバ  
ドメインの(原本)情報を管理するDNSサーバです。
- ◇ キャッシュサーバ  
クライアントに代わってコンテンツサーバに問合せを行うDNSサーバです。問合せた結果(③)を、複製(④)として一時的に記憶(キャッシュ)することから、キャッシュサーバと呼ばれています。また、クライアントに代わって問合せ(②)を行うことを「再帰動作」と呼びます。



DNSサービスは、インターネット上で稼動するホストの名前とIPアドレスを管理して名前解決を行う必要があります。効率的な名前解決が必要であることから、原本情報を管理するコンテンツサーバと、キャッシュと呼ぶ一時的な複製を管理するキャッシュサーバという2種類のサーバが連携しています。

### 用語定義

第3版から、RFC1035 (DNSの仕様を記載した文書) にあわせ、次のように用語を使用しています。

#### 再帰的な問合せ

DNSサーバに再帰動作を期待する  
クライアントからの問合せのことを示します。

#### 再帰動作

DNSサーバが、クライアントに代わって  
問合せを代行する動作を示します。

#### Stub Resolver

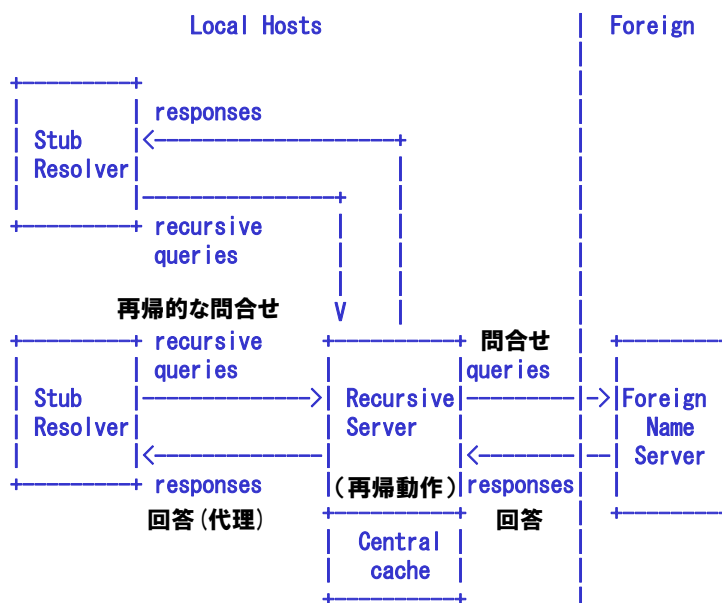
上図のクライアントに相当します。

#### Recursive Server

再帰動作を行なうDNSサーバのことであり、  
上図のキャッシュサーバに相当します。

#### Foreign Name Server

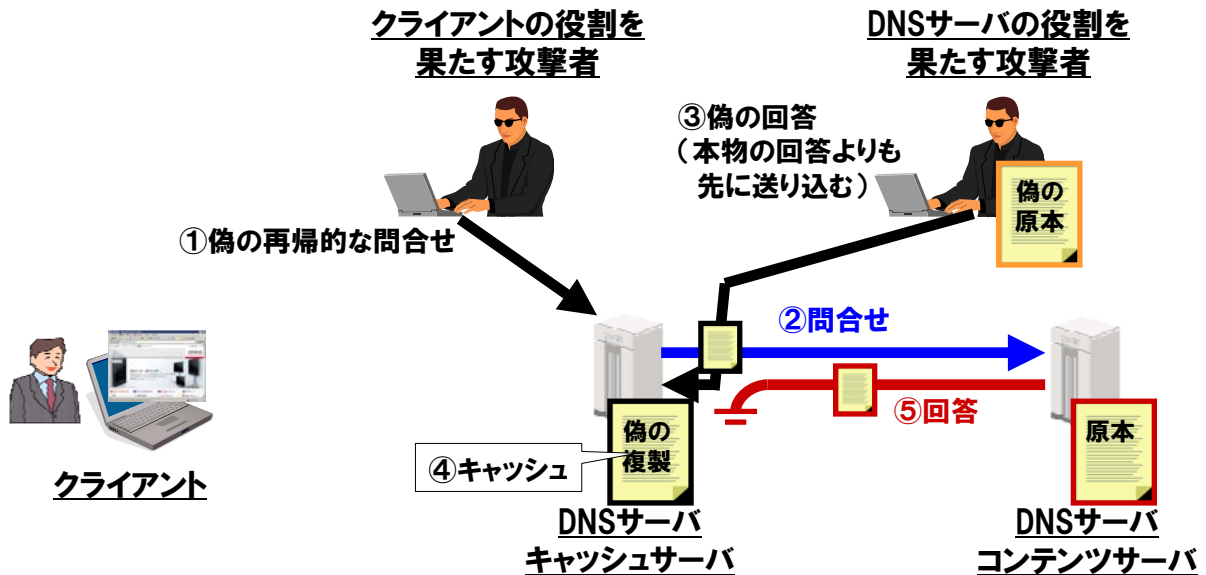
上図のコンテンツサーバに相当します。



<http://www.ietf.org/rfc/rfc1035.txt>

### DNSキャッシュポイズニングの実現手法

偽の再帰的な問合せ(①)に対して、本物のコンテンツサーバの回答(⑤)よりも先に偽の回答(③)を送り込むことで、キャッシュサーバに偽の情報(④)を覚えこませる(キャッシュさせる)攻撃です。

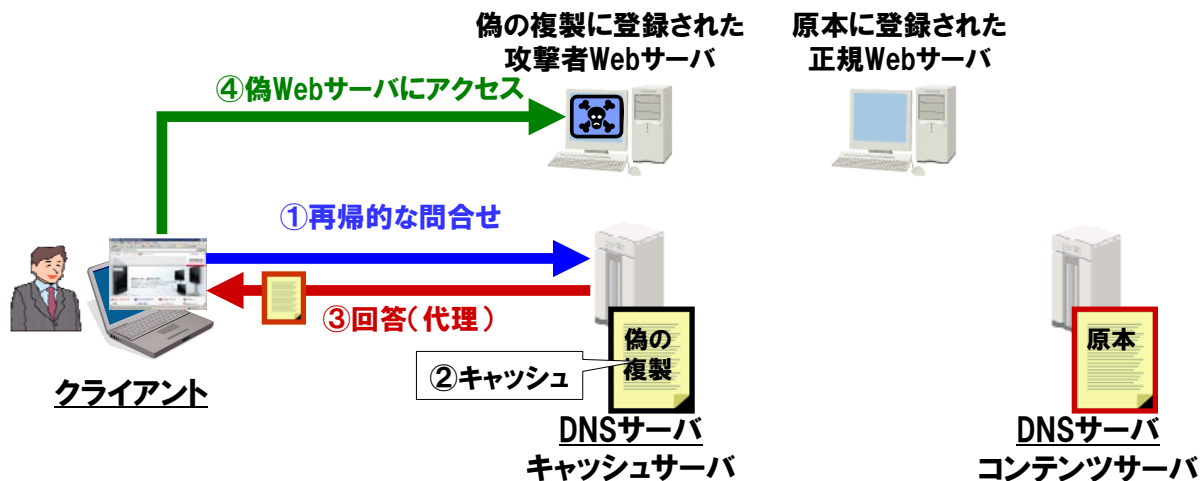


キャッシュポイズニングの実現手法を簡潔に説明すると、本物の回答よりも先に偽の回答をキャッシュサーバに送り込むことで、偽の情報を記憶させることです。このため、DNSキャッシュポイズニング対策では、偽の回答を送り込みにくい環境を整備し、キャッシュサーバが偽の情報を覚えられないという状況を作ることが重要となります。

### DNSキャッシュポイズニングによる脅威(その1)

【 攻撃者が罠をはったWebサーバへの誘導 】

キャッシュサーバは、クライアントの再帰的な問合せ(①)に対する回答(キャッシュされた偽の複製②)を持っている場合には、その複製を回答(③)します。結果として、クライアントは、提供された偽の情報により、攻撃者が罠をはったWebサーバ(④)に誘導されてしまうことになります。



DNSキャッシュポイズニングによるひとつめの脅威は、偽の情報を記憶したキャッシュサーバを利用してしまった場合、その偽の情報に誘導され、攻撃者が罠をはったWebサーバにアクセスしてしまうことです。

学校や企業の場合には、学校や企業で運用管理しているDNSサーバに対してキャッシュポイズニング対策をしておかないと、イントラネットからインターネット上のWebサーバを利用している多くの学生や社員が、偽の情報に誘導され、攻撃者が罠をはったWebサーバにアクセスする可能性を高めてしまいます。

脅威の説明については、次の資料を参考にしてください。

- DNSキャッシュポイズニングの脆弱性に関する注意喚起

[http://www.ipa.go.jp/security/vuln/documents/2008/200809\\_DNS.html](http://www.ipa.go.jp/security/vuln/documents/2008/200809_DNS.html)

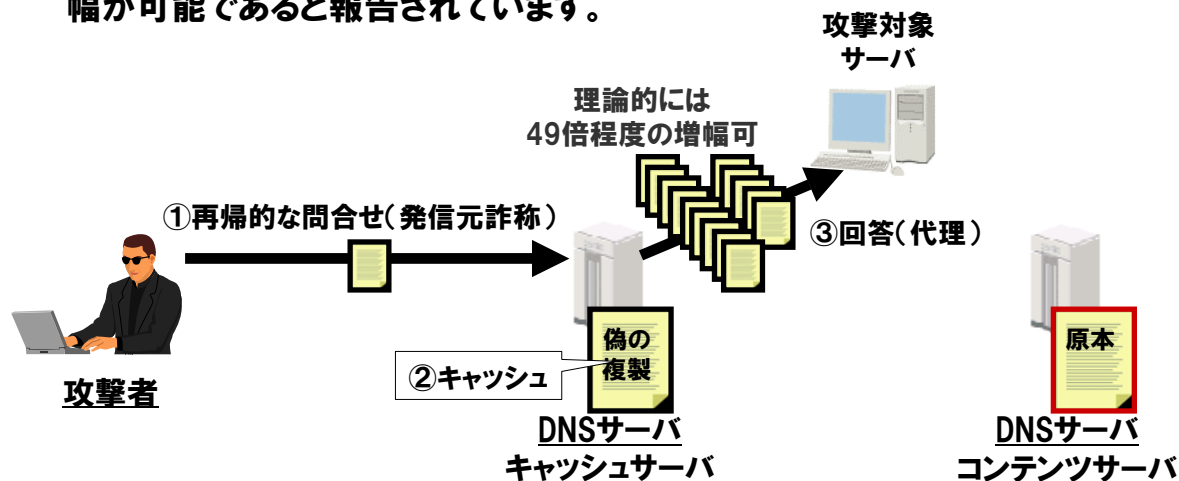
- DNSサーバの脆弱性に関する再度の注意喚起

[http://www.ipa.go.jp/security/vuln/documents/2008/200812\\_DNS.html](http://www.ipa.go.jp/security/vuln/documents/2008/200812_DNS.html)

### DNSキャッシュポイズニングによる脅威(その2)

【 DoS攻撃ための増幅装置 】

データサイズの大きな偽の複製(②)を覚えこませた(キャッシュさせた)後、キャッシュサーバに対して、発信元を詐称した再帰的な問合せ(①)を行います。結果として、データサイズの大きな偽の複製(③)を、攻撃対象サーバに送信してしまうことになります。理論的には49倍程度のトラフィック増幅が可能であると報告されています。



DNSキャッシュポイズニングによるふたつめの脅威は、DoS(Denial of Service)攻撃のための増幅装置として利用されてしまう可能性があることです。

ふたつめの脅威を悪用した活動は、1999年にオーストラリアのCSIRT(Computer Security Incident Response Team)から報告されています。また、2006年には、JPCERT/CC、@policeから「DNSの再帰的な問合せを使ったDDoS攻撃」について報告されています。

DNSの再帰的な問合せを使ったDDoS攻撃については、次の資料を参考にしてください。

- DNSの再帰的な問い合わせを悪用したDDoS攻撃手法の検証について  
[http://www.cyberpolice.go.jp/server/rd\\_env/pdf/20060711\\_DNS-DDoS.pdf](http://www.cyberpolice.go.jp/server/rd_env/pdf/20060711_DNS-DDoS.pdf)
- AL-1999.004 -- Denial of Service (DoS) attacks using the Domain Name System (DNS)  
<http://www.auscert.org.au/render.html?it=80>

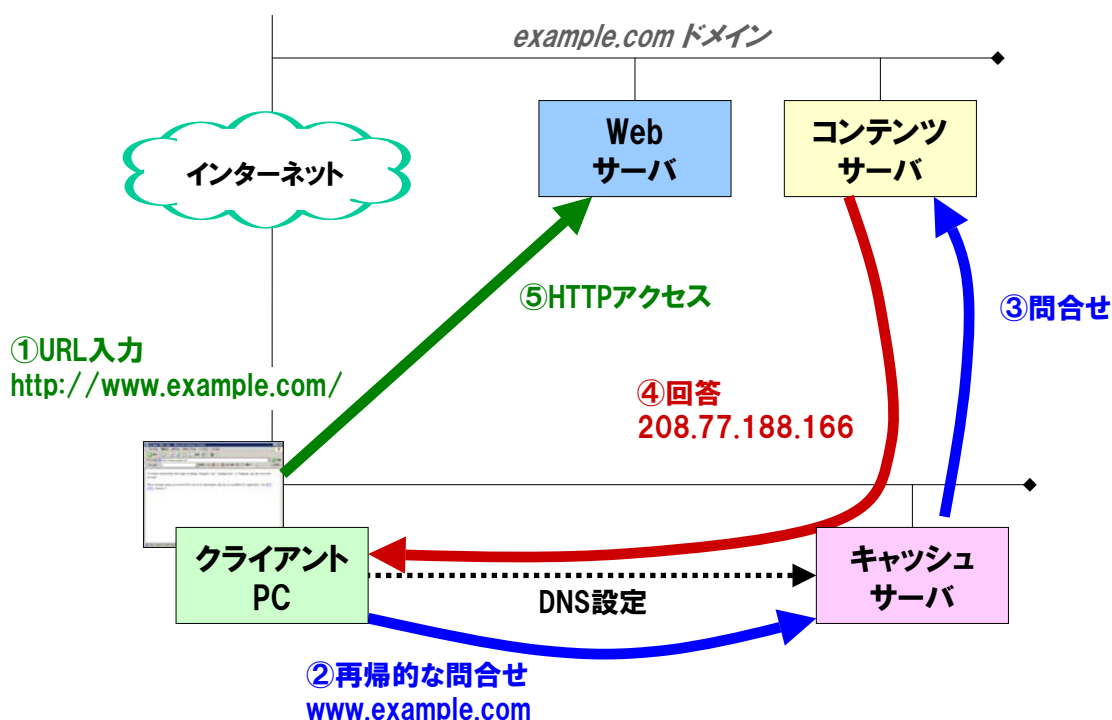
## 2. DNSの動作と関連ツール

- 2.1 DNSの動作概説
- 2.2 whoisサービス
- 2.3 nslookupコマンド
- 2.4 まとめ

「2. DNSの動作と関連ツール」では、DNSの問合せ動作と、その関連ツールであるwhoisサービスとnslookupの使い方を説明します。

## 2.1 DNSの動作概説

インターネット直接接続PCの場合(ブラウザでプロキシ設定なし)



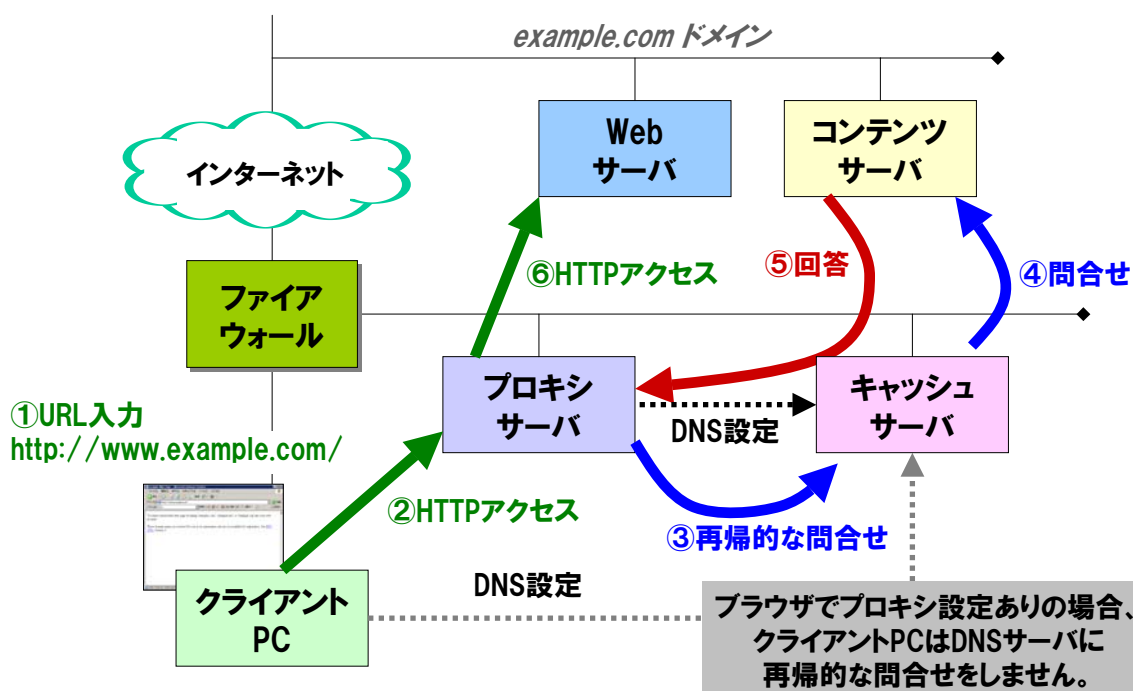
通常、Webサーバへアクセスする際にDNSがどのように使用されているかを説明します。

インターネットに直接接続されているPCの場合

- ① ブラウザへURL (`http://www.example.com/`) を入力すると、
- ② クライアントPCは、OSで設定されているDNS設定(キャッシュサーバ)に対して、`www.example.com` の名前解決を要求(再帰的な問合せ)します。
- ③ キャッシュサーバは、`example.com` ドメインを管理しているコンテンツサーバに問合せを行い、
- ④ コンテンツサーバからの回答(`208.77.188.166`)をキャッシュサーバ経由でクライアントPCに転送します。
- ⑤ クライアントPCのブラウザは、回答(`208.77.188.166`)で示されたWebサーバにHTTPアクセスを行います。

## 2.1 DNSの動作概説

### イントラネット接続PCの場合(ブラウザでプロキシ設定あり)



イントラネットに接続されているPCの場合、一般的にはブラウザの設定で、プロキシサーバが指定されています。

#### ブラウザでプロキシ設定のあるPCの場合

- ① ブラウザへURL (`http://www.example.com/`) を入力すると、
- ② クライアントPCのブラウザはプロキシサーバにHTTPアクセスを行います。
- ③ プロキシサーバは、設定されているDNS設定(キャッシュサーバ)に対して、`www.example.com`の名前解決を要求(再帰的な問合せ)します。
- ④ キャッシュサーバは、`example.com`ドメインを管理しているコンテンツサーバへ問合せを行い、
- ⑤ コンテンツサーバからの回答(`208.77.188.166`)をキャッシュサーバ経由でプロキシサーバに転送します。
- ⑥ プロキシサーバは、回答(`208.77.188.166`)で示されたWebサーバにHTTPアクセスを行います。

ブラウザにプロキシの設定がある場合は、クライアントPCはDNS設定を参照せず、プロキシサーバがDNS設定を参照して名前解決を行います。ここが前述のインターネット直接接続PCの場合と異なる点です。

### whoisサービスとは・・・

IPアドレスやドメイン名の登録者などに関する情報を、インターネットユーザが誰でも参照できるサービスです。このサービスを使用することで、ドメインの(原本)情報管理を行うDNSサーバ(コンテンツサーバ)を確認できます。whoisサービスは、トップレベルドメイン別にサービスサイトが存在します。トップレベルドメインとは、「.jp」、「.com」、「.net」などを示します。

◇ JPRS whois ( .jp の場合 )

<http://whois.jprs.jp/>

◇ InterNIC WHOIS( .com、.net などの場合 )

<http://www.internic.net/whois.html>

その他、以下の whois サービスサイトがあります。

APNIC WHOIS、AfriNIC WHOIS、ARIN WHOIS、RIPE NCC WHOIS、LACNIC WHOIS

IPアドレスやドメイン名の登録者などドメイン関連する情報を参照できるwhoisサービスについて紹介します。whois サービスに掲載されている情報は、ドメイン登録時／更新時に申請した情報です。

トップレベルドメインは、分野別トップレベルドメインである gTLD(generic Top Level Domain)と国コードトップレベルドメインである ccTLD(country code Top Level Domain)に分けられます。interNIC WHOIS のサービスサイトで gTLD である「.com」や「.net」の情報が検索でき、JPRSのwhoisサービスサイトで、「jp」の情報は検索できます。「.jp」は ccTLDに属します。JPRS以外に ccTLDのサービスサイトはそれぞれの地域に存在します。

- アジア太平洋地域:APNIC WHOIS
- アフリカ地域:AfriNIC WHOIS
- 北米地域:ARIN WHOIS
- 欧州・アフリカ地域:RIPE NCC WHOIS
- 南米カリブ海地域:LACNIC WHOIS

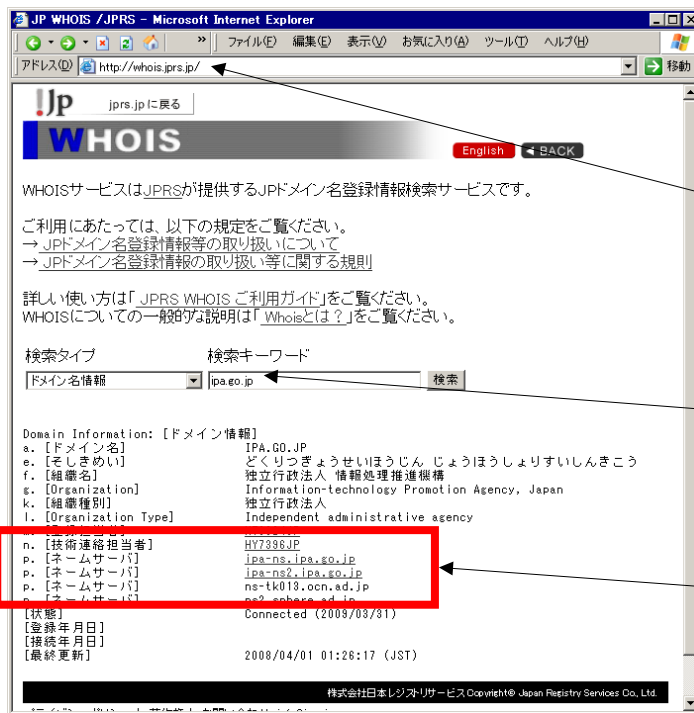
トップレベルドメインの詳細については次の資料を参照してください。

● ドメイン名の種類

<http://www.nic.ad.jp/ja/dom/types.html>

## 2.2 ドメイン(原本)情報管理サーバの確認方法

### .jp ドメインの whois サービスサイト



① whois サービスにアクセス  
<http://whois.jprs.jp/>

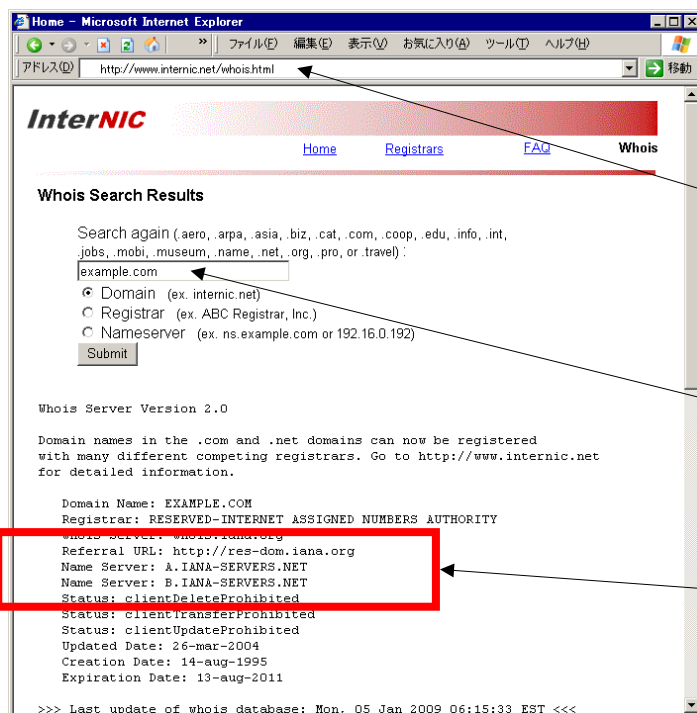
② ドメイン名を入力  
ipa.go.jp

③ 結果  
ipa.go.jpドメインの  
(原本)情報管理をしている  
DNSサーバ(コンテンツサーバ)

JPDメイン用のwhoisサービスの使い方です。Webベースのツールですので、ブラウザから「<http://whois.jprs.jp/>」にアクセスして、対象のドメイン名を入力後、「検索」ボタンをクリックしてください。

## 2.2 ドメイン(原本)情報管理サーバの確認方法

### .com の whois サービスサイト



① whois サービスにアクセス  
<http://www.internic.net/whois.html>

② ドメイン名を入力  
example.com

③ 結果  
example.comドメインの  
(原本)情報管理をしている  
DNSサーバ(コンテンツサーバ)

.comドメイン用のwhoisサービスの使い方です。Webベースのツールですので、ブラウザから「<http://www.internic.net/whois.html>」にアクセスして、対象のドメイン名を入力後、「Submit」ボタンをクリックしてください。

## 2.3 nslookup コマンド

nslookupは、DNSサーバに登録されている情報を参照するコマンドです。

参照可能な情報の例

- ◇ ホスト名からIPアドレス
- ◇ IPアドレスからホスト名
- ◇ ドメイン(原本)を管理しているDNSサーバ
- ◇ ドメインのメールサーバ

### クライアントPCにおける「DNS設定」の確認

```
C:¥>
C:¥>ipconfig /all

Ethernet adapter ローカル エリア接続:

    Connection-specific DNS Suffix  . : 
    Description . . . . . : Intel(R) PRO/1000

    Physical Address. . . . . : 00-xx-xx-xx-xx-xx
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 10.10.10.200
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.10.10.251
    DNS Servers . . . . . : 10.10.10.10
                           10.10.10.20
```

### ホスト名からIPアドレスの検索

```
C:¥>
C:¥>nslookup www.example.com.
Server: dns-internal.ipa.go.jp
Address: 10.10.10.10

Non-authoritative answer:
Name: www.example.com
Address: 208.77.188.166
```

クライアントPCで設定しているDNSサーバの名前とアドレス

クライアントPCにおける「DNS設定」の確認と、DNSサーバに登録されている情報を参照するコマンドであるnslookupについて紹介します。

Windowsの場合、コマンドプロンプトを開いた後、ipconfig を入力すると、左側の「クライアントPCにおける「DNS設定」の確認」で示す通り、現在設定されている DNS サーバを確認できます。DHCPでネットワーク設定が自動設定されている場合も確認できます。

nslookupは、コマンドに引き続いて「ホスト名」を入力することでIPアドレスを参照できます。右側の「ホスト名からIPアドレスの検索」は、nslookupを用いて、www.example.com. のIPアドレスを参照した例です。

## 2.3 nslookupコマンド

nslookupコマンドの表示結果の例を、環境別に説明します。

### ◇ 使用コマンド書式

nslookup [検索するホスト名] [使用するDNSサーバ]

[使用するDNSサーバ]を指定しない場合には、クライアントPCにおける「DNS設定」が使用されます。

接続環境	使用するDNSサーバ	パターン
インターネット直接接続PC	アクセス制限なし	①
	アクセス制限あり	②
イントラネット接続PC	組織内キャッシュサーバ	③
	組織外キャッシュサーバ	④

nslookupは、コマンドに引き続いて「検索するホスト名」「使用するDNSサーバ」を入力することで、「使用するDNSサーバ」に対して、「検索するホスト名」の名前解決を依頼できます。

例: DNSサーバ(ipa-ns.ipa.go.jp)に、www.example.comを問合せる場合  
nslookup www.example.com. ipa-ns.ipa.go.jp

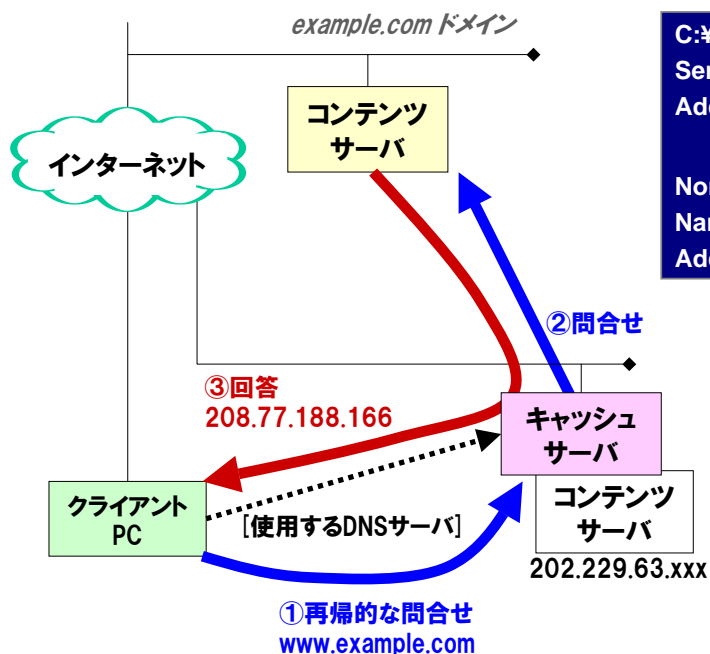
また、[使用するDNSサーバ]を指定しない場合には、クライアントPCの「DNS設定」で指定されているDNSサーバに対して名前解決を依頼します。なお、www.example.com.の右端に「ドット」がついているのは、名前が終端していることを意味します。

例: クライアントPCの「DNS設定」で指定されているDNSサーバに、www.example.comを問合せる場合  
nslookup www.example.com.

## 2.3 nslookupコマンド①

### インターネット直接接続PCの場合

[使用するDNSサーバ]としてインターネット上のアクセス制限 **されていない** キャッシュサーバ(含むコンテンツサーバ兼用)を指定した場合



```
C:\>nslookup www.example.com. 202.229.63.xxx
Server: aaa.aaa.xxx
Address: 202.229.63.xxx

Non-authoritative answer:
Name:   www.example.com
Address: 208.77.188.166
```

クライアントPCからの再帰的な問合せの回答が、キャッシュサーバを経由した回答の場合には、

Non-authoritative answer:

と表示されます。これは、コンテンツサーバからのオリジナルの回答ではないことを意味します。

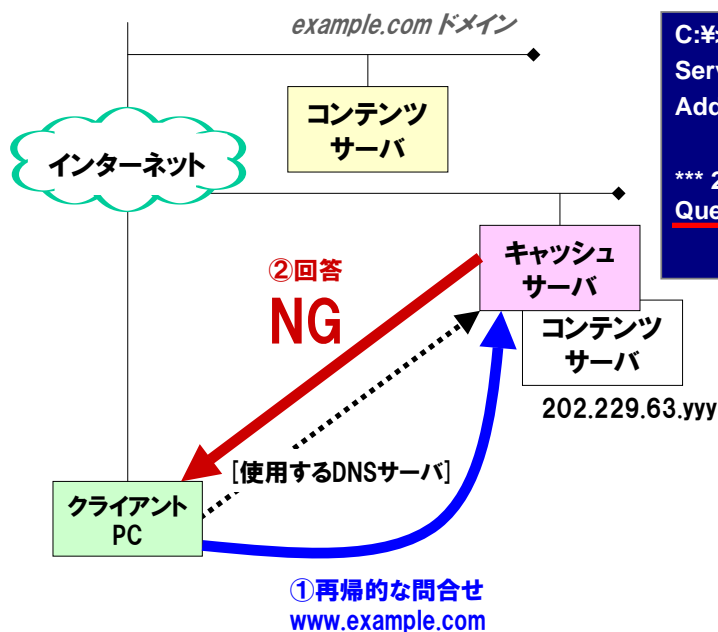
### パターン①

インターネットに直接接続しているPCが、インターネット上のアクセス制限されていないキャッシュサーバ(含むコンテンツサーバ兼用)を指定した場合のnslookupコマンド実行結果の例です。

## 2.3 nslookupコマンド②

### インターネット直接接続PCの場合

[使用するDNSサーバ]としてインターネット上のアクセス制限 **されている** キャッシュサーバ(含むコンテンツサーバ兼用)を指定した場合



```
C:\>nslookup www.example.com. 202.229.63.yyy
Server: aaa.aaa.yyy
Address: 202.229.63.yyy

*** 202.229.63.yyy can't find www.example.com.:
Query refused
```

クライアントPCにキャッシュサーバへのアクセス許可権限がない場合には、再帰的な問合せは拒否されます。

### パターン②

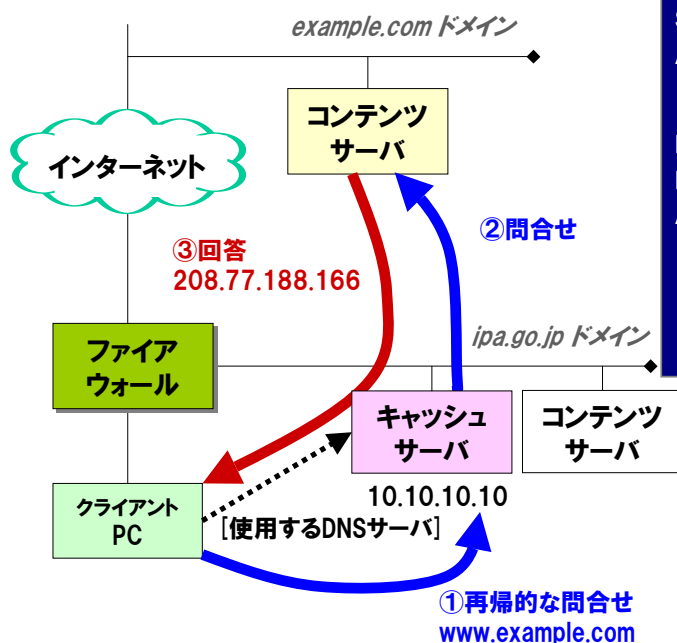
インターネットに直接接続しているPCが、インターネット上のアクセス制限されているキャッシュサーバ(含むコンテンツサーバ兼用)を指定した場合のnslookupコマンド実行結果の例です。

クライアントPCがキャッシュサーバへのアクセス許可権限がない(BIND の設定:allow-query { address\_match\_list }; を用いた)場合には、再帰的な問合せを受け付けないため、キャッシュサーバからの回答が「Query refused」という拒否された内容となります。

## 2.3 nslookupコマンド③

### イントラネット接続PCの場合

[使用するDNSサーバ]として自組織のキャッシュサーバを指定した場合



```
C:\>nslookup www.example.com. 10.10.10.10
Server: dns-internal.ipa.go.jp
Address: 10.10.10.10
```

```
Non-authoritative answer:
Name: www.example.com
Address: 208.77.188.166
```

nslookupコマンド①と同じ結果が得られます。

### パターン③

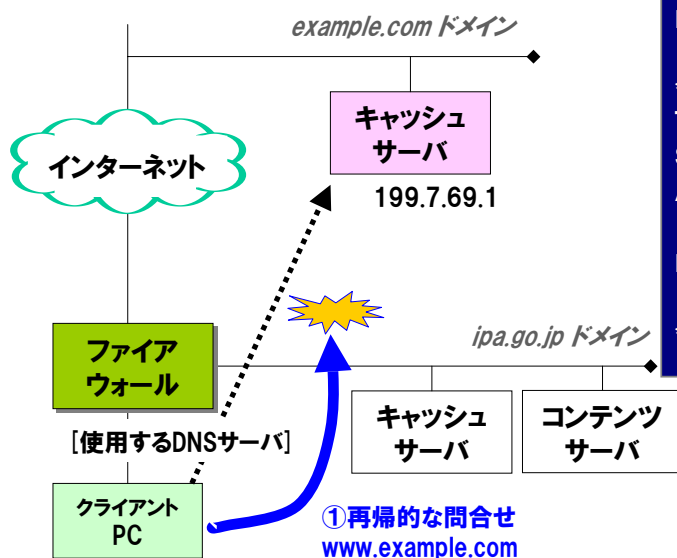
イントラネットに接続されたPCが、自組織のキャッシュサーバを指定した場合のnslookupコマンド実行結果の例です。

実行結果は、パターン①と同じです。クライアントPCからの再帰的な問合せが、キャッシュサーバを経由した回答の場合には、「Non-authoritative answer」と表示されます。これは、コンテンツサーバからのオリジナルの回答ではないことを意味します。

## 2.3 nslookupコマンド④

### イントラネット接続PCの場合

[使用するDNSサーバ]としてインターネット上のキャッシュサーバを指定した場合



```
C:\>nslookup www.example.com. 199.7.69.1
DNS request timed out.
  timeout was 2 seconds.
*** Can't find server name for address 199.7.69.1:
Timed out
Server: UnKnown
Address: 199.7.69.1

DNS request timed out.
  timeout was 2 seconds.
*** Request to UnKnown timed-out
```

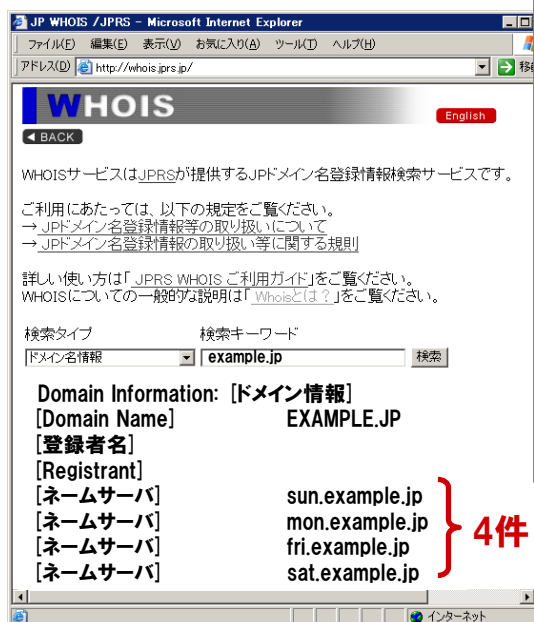
ファイアウォールなどで、[使用するDNSサーバ]へのアクセスが制限されている場合には、タイムアウトが発生します。

#### パターン④

イントラネットに接続されたPCがインターネット上のキャッシュサーバを指定した場合のnslookupコマンド実行結果の例です。

クライアントPCからインターネット上のキャッシュサーバへの再帰的な問合せがファイアウォールやルータなどでアクセス制限がされている場合、タイムアウトが発生します。

## whoisサービスに登録されているDNSサーバ(コンテンツサーバ)と、nslookupで表示されるDNSサーバ(コンテンツサーバ)は一致していますか？



```
C:\>nslookup -q=NS example.jp.  
Server: dns-internal.ipa.go.jp  
Address: 10.10.10.10
```

Non-authoritative answer:

```
example.jp nameserver = wed.example.jp  
example.jp nameserver = sun.example.jp  
example.jp nameserver = mon.example.jp  
example.jp nameserver = fri.example.jp  
example.jp nameserver = sat.example.jp
```

5件

```
wed.example.jp internet address = 202.229.xxx.5  
sun.example.jp internet address = 202.229.xxx.1  
mon.example.jp internet address = 202.229.xxx.2  
fri.example.jp internet address = 202.229.xxx.3  
sat.example.jp internet address = 202.229.xxx.4
```

ドメイン名の登録と DNS サーバの設定に関する注意喚起  
[http://www.ipa.go.jp/security/vuln/20050627\\_dns.html](http://www.ipa.go.jp/security/vuln/20050627_dns.html)

DNSの問合せ動作と、その関連ツールであるwhoisサービスとnslookupの使い方を説明しました。

まとめでは、whoisサービスとnslookupを使った、DNSサーバ(コンテンツサーバ)の一致について補足したいと思います。

whois サービスに掲載されている情報は、ドメイン登録時／更新時に申請した情報です。一方、nslookupコマンドで参照できる情報は、実際に稼動しているコンテンツサーバに登録されている情報です。

この事例(example.jp)の場合、whois に登録されているネームサーバ(=DNSサーバ)は4件で、nslookupで参照できたDNSサーバは5件です。これは、ドメイン登録時／更新時に申請した内容と実際に稼動しているコンテンツサーバに登録されている情報とに乖離があることを意味します。一般的には、whoisサービスに登録されている情報とnslookupにて参照できる情報が同じであることが推奨されています。この機会にwhoisサービスに登録されている情報とnslookupにて参照できる情報(実際に管理している情報)が一致しているかを確認し、整合性のとれた運用管理をしましょう。

詳細については、次の資料を参照してください。

- ドメイン名の登録とDNSサーバの設定に関する注意喚起

[http://www.ipa.go.jp/security/vuln/20050627\\_dns.html](http://www.ipa.go.jp/security/vuln/20050627_dns.html)

## 3. 検査ツールの使い方と注意点

### 3.1 Cross-Pollination Check

### 3.2 DNS-OARC Randomness Test (Web版)

### 3.3 DNS-OARC Randomness Test (コマンドライン版)

### 3.4 まとめ

「3. 検査ツールの使い方と注意点」では、DNSサーバに対してキャッシュポイズニング対策の検査ができるツールの使い方と注意点を説明します。

- Cross-Pollination Check
- DNS-OARC Randomness Test (Web版)
- DNS-OARC Randomness Test (コマンドライン版)

事前に、次に示す用語の意味を理解しておく必要があります。

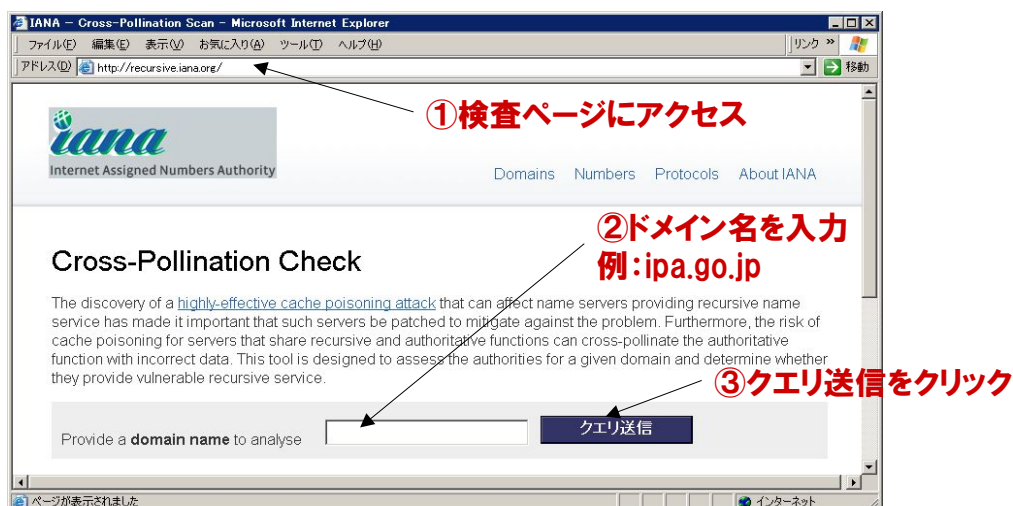
- コンテンツサーバ
- キャッシュサーバ
- 再帰的な問合せ
- nslookup

## 3.1 Cross-Pollination Checkの使い方(1/2)

<http://recursive.iana.org/>

コンテンツサーバを検査するツールです。ドメイン名を入力すると、そのドメインのコンテンツサーバを調べて、1)再帰動作の可否、2)送信元ポート番号のランダム性有無を確認し、結果を表示してくれます。

自分の管理しているドメイン名を入力することで、自ドメインのコンテンツサーバを検査可能です。

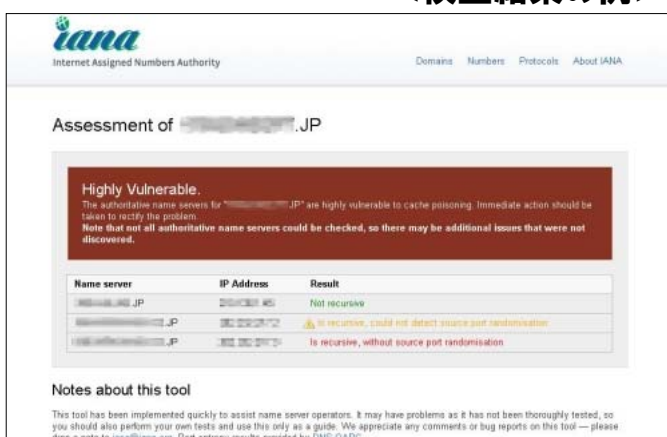


コンテンツサーバを検査するツール「Cross-Pollination Check」の使い方です。Webベースのツールですので、ブラウザから「<http://recursive.iana.org/>」にアクセスして、検査対象のドメイン名を入力後、「クエリ送信」ボタンをクリックしてください。

### 3.1 Cross-Pollination Checkの使い方(2/2)

コンテンツサーバ毎に結果が表示されます。結果は、Highly Vulnerable、Vulnerable、Safeのいずれかになります。  
Safeの場合は、再帰的な問合せに回答しなかったことを示しており、再帰動作が無効化もしくは制限されていることがわかります。  
Safe以外の場合は、再帰的な問合せに回答したことを示しており、セキュリティ修正プログラム(ポート番号のランダム化)が適用されていればVulnerable、されていなければHighly Vulnerableになります。

#### <検査結果の例>



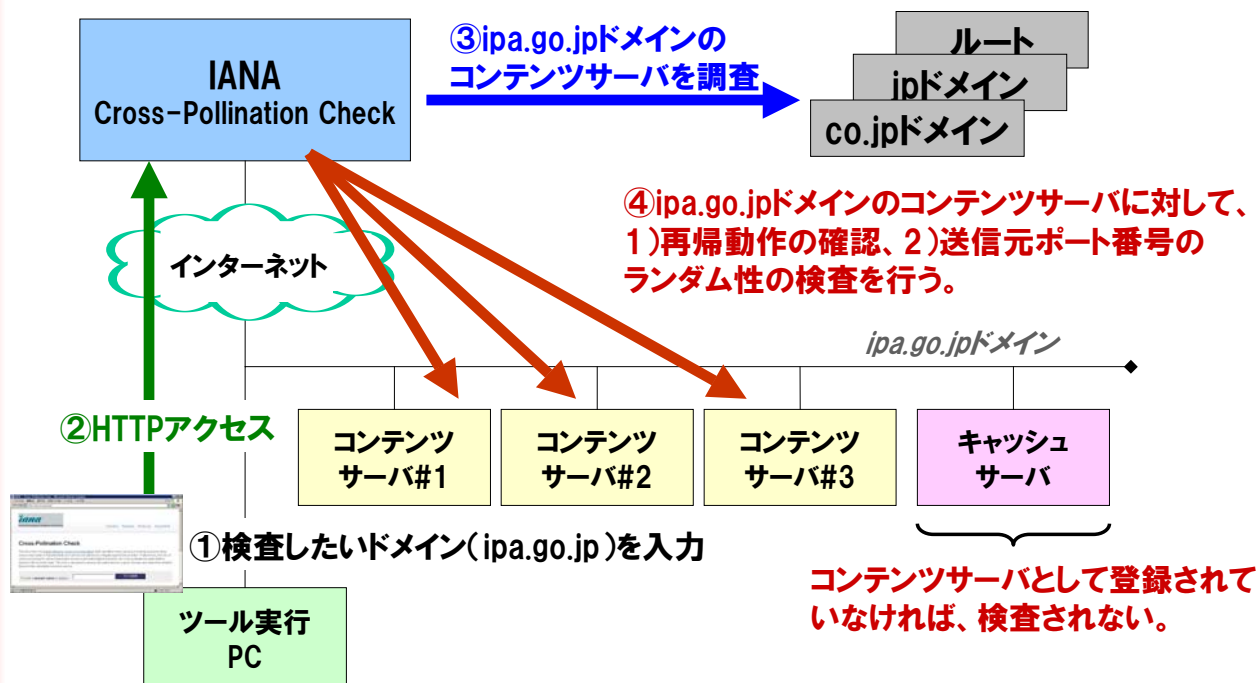
#### <検査結果の見方>

	再帰動作無効	送信元ポートのランダム化
Highly Vulnerable	No	No
Vulnerable	No	Yes
Safe	Yes	-

「Cross-Pollination Check」の検査結果の見方は、上記の通りです。結果は、Highly Vulnerable、Vulnerable、Safeの3つのうち、いずれかになります。

### 3.1 Cross-Pollination Checkの仕組みと注意点

インターネット側から再帰的な問合せを送信し、回答があれば、送信元ポート番号のランダム性を検査します。



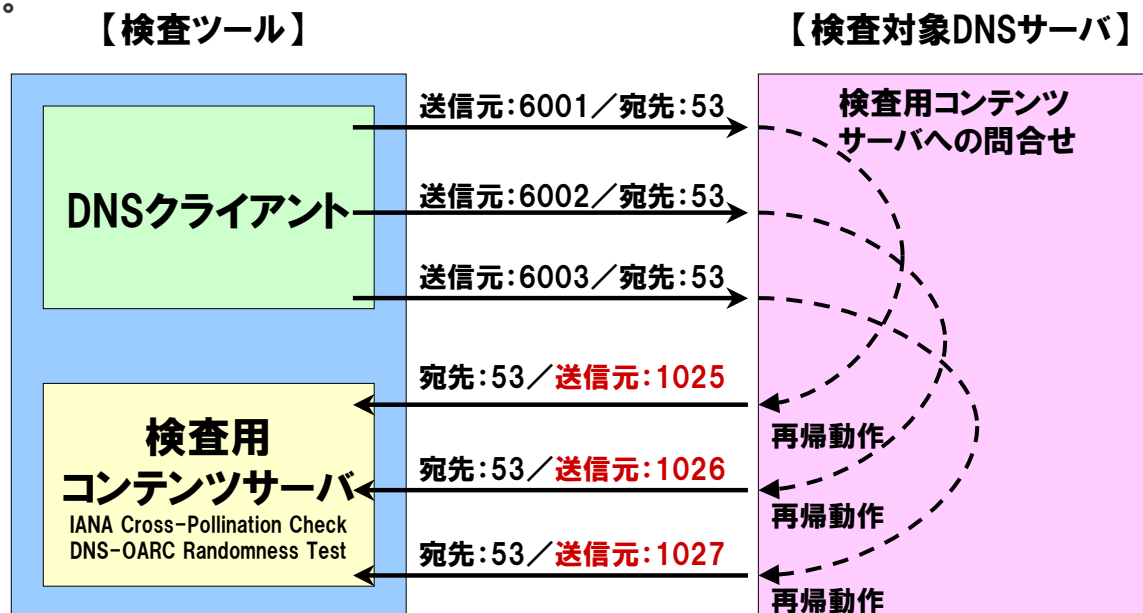
「Cross-Pollination Check」が実施する検査の仕組みについて説明します。

ここでのポイントは、

- 1)インターネットからの再帰的な問合せに回答するDNSサーバに対してのみ、セキュリティ修正プログラム適用有無の検査ができるという点と、
- 2)コンテンツサーバとして登録されていないDNSサーバ(キャッシュサーバ)は検査対象とならない点です。

### 3.1 検査ツールが送信元ポート番号を確認する仕組み

検査ツールは、自身のコンテンツサーバへ問合せが発生するように、検査対象に再帰的な問合せを送信することで、再帰動作時の送信元ポート番号を調査します。また、複数回の問合せを行うことで、ポート番号のランダム性を検査しています。



検査ツールでは、送信元ポート番号のランダム性を検査します。

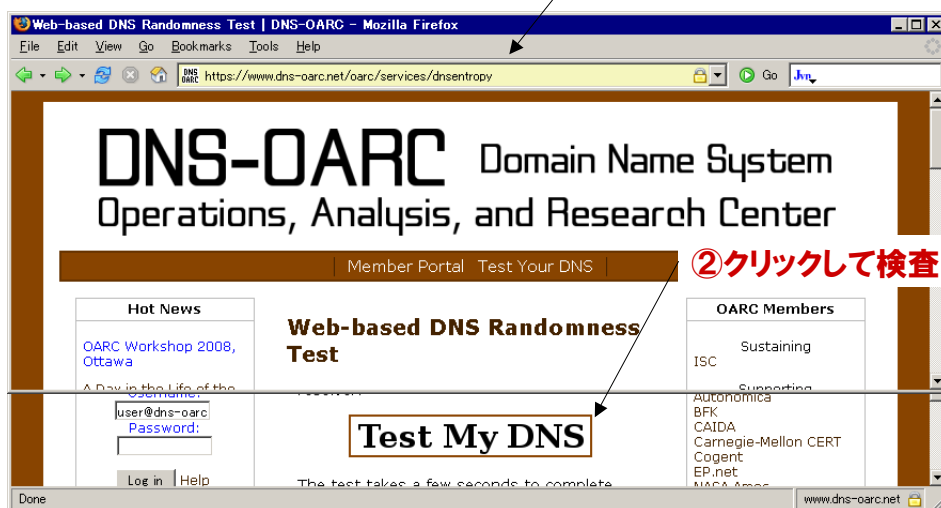
上記では、検査ツールがどのようにして検査対象DNSサーバの送信元ポート番号と、そのランダム性について確認しているかを補足説明しています。

## 3.2 DNS-OARC(Web版)の使い方(1/2)

<https://www.dns-oarc.net/oarc/services/dnsentropy>

自分が使用しているキャッシュサーバを検査するツールです。「Test My DNS」ボタンをクリックすると、自動的に検査が開始され、1)送信元ポート番号のランダム性有無、2)トランザクションIDのランダム性有無の検査結果が新しいウィンドウに表示されます。

①検査ページにアクセス



②クリックして検査開始

前述した通り、「Cross-Pollination Check」では、コンテンツサーバでないDNSサーバを検査できません。このため、キャッシュサーバを検査する場合は、「DNS-OARC Randomness Test」を使用する必要があります。このツールには、Web版とコマンドライン版があります。

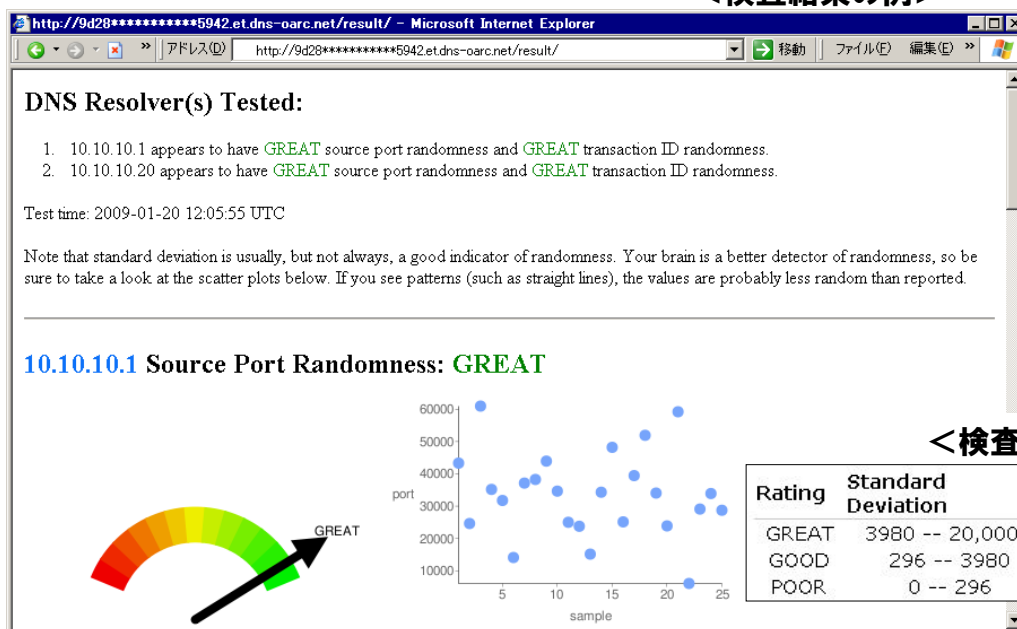
まずWeb版について説明します。

「<https://www.dns-oarc.net/oarc/services/dnsentropy>」にアクセスし、「Test My DNS」ボタンをクリックすると、検査が開始されます。なお、このボタンは、PCの環境によっては、黒く塗り潰された画像として表示される場合があります。

## 3.2 DNS-OARC(Web版)の使い方(2/2)

キャッシュサーバ毎に送信元ポート番号及びトランザクションIDのランダム性が GREAT、GOOD、POORの3段階で表示されます。**注)必ずしも全てのキャッシュサーバが検査対象になる訳ではありません。**

<検査結果の例>



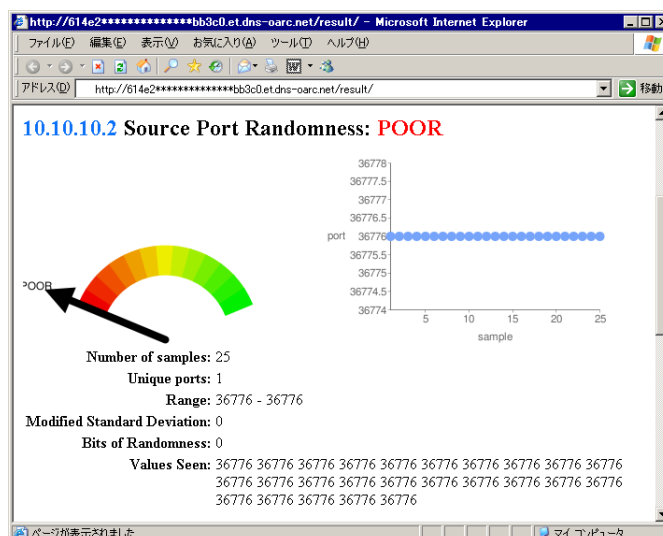
<検査結果の見方>

「DNS-OARC Randomness Test(Web版)」の検査結果は上記の通りです。画面上部に結果のサマリが表示され、GREAT、GOOD、POORの3段階で評価されます。以降、結果の詳細がグラフとともに表示されます。ここでの注意点は、設定している全てのキャッシュサーバが検査対象となる訳ではありませんので、意図したDNSサーバが全て検査されているかを確認する必要があります。

GREAT、GOOD、POORの評価は、送信元ポート番号及びトランザクションIDの値が16ビットをどの程度まんべんなく使用しているかに基づいています。なお、この2つの値が推測できるということは、キャッシュポイズニング攻撃が実現しやすいことを意味しています。

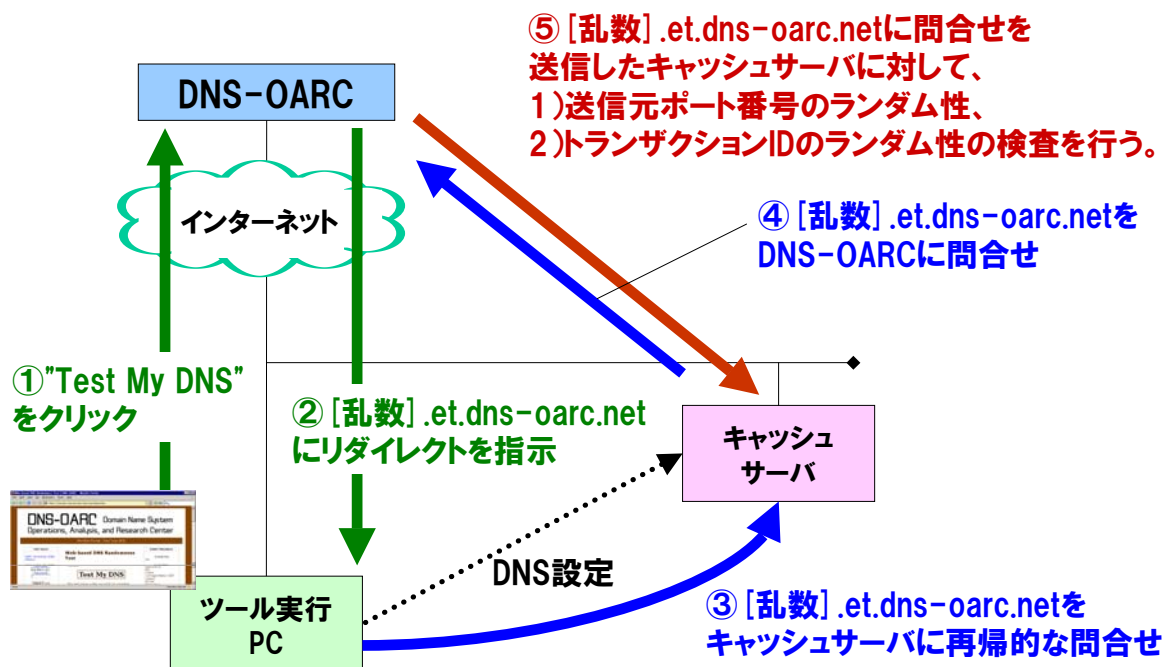
右図は、送信元ポート番号のランダム性POORと判定された事例です。値にばらつきがないと、次に発生しうる値を容易に類推できます。

DNSキャッシュポイズニング対策においては、送信元ポート番号及びトランザクションIDの値を推測されにくい状況、すなわち、POORと評価されないこと、**GOODよりは、GREATと評価されるような状況を作り出すことが重要です。**



## 3.2 DNS-OARC(Web版)の仕組み

Webアクセス時に名前解決を行ったキャッシュサーバが検査対象となります。  
すなわち、OSに設定しているDNSサーバ(=DNS設定)です。

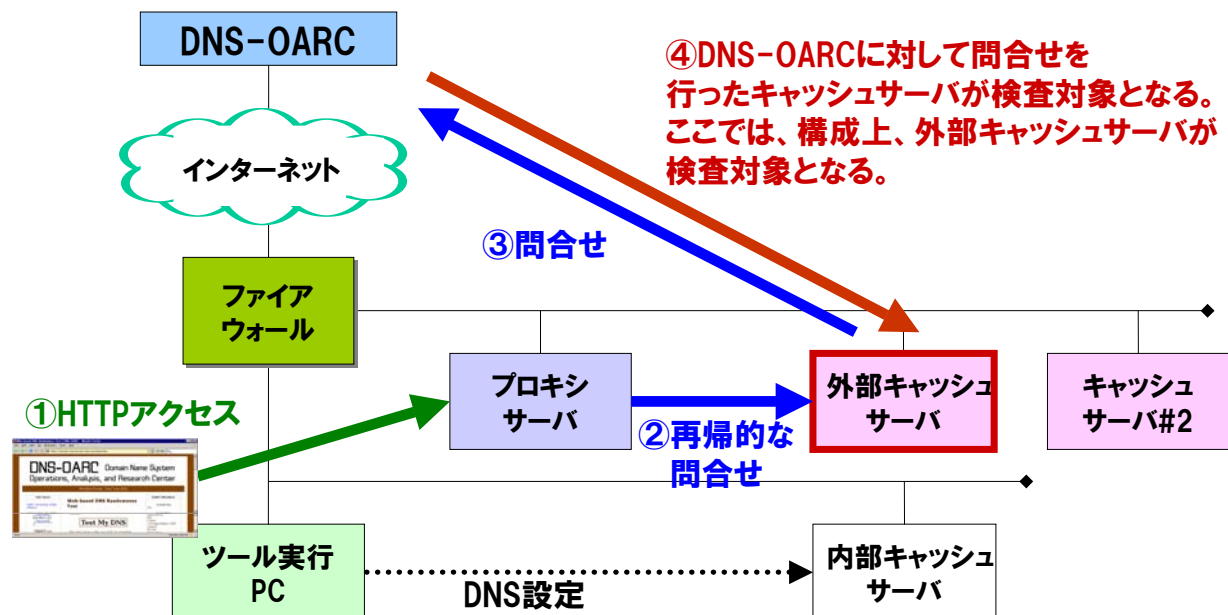


上記は、インターネットに直接接続しているPCで「DNS-OARC Randomness Test(Web版)」を実行した場合の検査の流れを示しています。

ツール実行PC上に設定しているDNSサーバ(=DNS設定の参照するキャッシュサーバ)がDNS-OARCのコンテンツサーバに問合せを行うため、そのDNSサーバが検査対象となります。

### 3.2 DNS-OARC(Web版)の注意点

プロキシ環境においては、プロキシに設定されているキャッシュサーバが検査対象となるため、必ずしもツール実行PCに設定されたキャッシュサーバが検査対象となる訳ではありません。



イントラネットに接続したPCの場合、Webサーバのアクセスにはプロキシサーバ経由であることが一般的です。この場合は、ツール実行PC自身がDNSサーバに名前解決を行うのではなく、プロキシサーバがDNSサーバに名前解決を行います。したがって、ツール実行PCに設定しているDNSサーバが検査対象になる訳ではありません。

また、この場合、プロキシサーバが使用しないDNSサーバ(上記の場合「キャッシュサーバ#2」)は検査対象になりません。検査したい場合は、コマンドライン版を使用する必要があります。

### 3.3 DNS-OARC(コマンドライン版)の使い方(1/2)



DNS-OARCには、Web版の他にコマンドライン版があります。コマンドライン版では、検査対象のDNSサーバを任意に指定可能です。

コマンドライン版は、WindowsもしくはLinuxのnslookupコマンドで実行します(digコマンドでも同様に実行できます。詳しくは下記解説ページを参照ください)。

◇ 送信元ポート番号のランダム性検査の場合

nslookup -q=TXT -timeout=10 porttest.dns-oarc.net. [検査対象DNSサーバ]

◇ トランザクションIDのランダム性検査の場合

nslookup -q=TXT -timeout=10 txidtest.dns-oarc.net. [検査対象DNSサーバ]

解説ページ

<https://www.dns-oarc.net/oarc/services/porttest>

<https://www.dns-oarc.net/oarc/services/txidtest>

<http://itpro.nikkeibp.co.jp/article/COLUMN/20080811/312660/>

「DNS-OARC Randomness Test(コマンドライン版)」では、OSに標準搭載されているnslookupコマンドを使用します。なお、nslookupコマンド以外に、digコマンドを使用することも可能です。なお、コマンドライン版の場合は、送信元ポート番号とトランザクションIDは別々に検査する必要があります。

使い方については、次の資料を参照してください。

- porttest.dns-oarc.net -- Check your resolver's source port behavior  
<https://www.dns-oarc.net/oarc/services/porttest>
- txidtest.dns-oarc.net -- Check your resolver's transaction ID behavior  
<https://www.dns-oarc.net/oarc/services/txidtest>
- 詳細が明かされたDNSキャッシュ・ポイズニングの新手法  
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080811/312660/>

### 3.3 DNS-OARC(コマンドライン版)の使い方(2/2)

検査対象「ns.example.com(192.168.0.2)」の実行結果は次の通りです。

```
C:\>nslookup -q=TXT -timeout=10 porttest.dns-oarc.net. ns.example.com.  
Server: ns.example.com  
Address: 192.168.0.2
```

Non-authoritative answer:

```
porttest.dns-oarc.net canonical name =porttest.y.x.w.v.u.t.s.r.q.p.o.n.m.l.k.  
j.i.h.g.f.e.d.c.b.a.pt.dns-oarc.net  
porttest.y.x.w.v.u.t.s.r.q.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.pt.dns-oarc.net  
text =
```

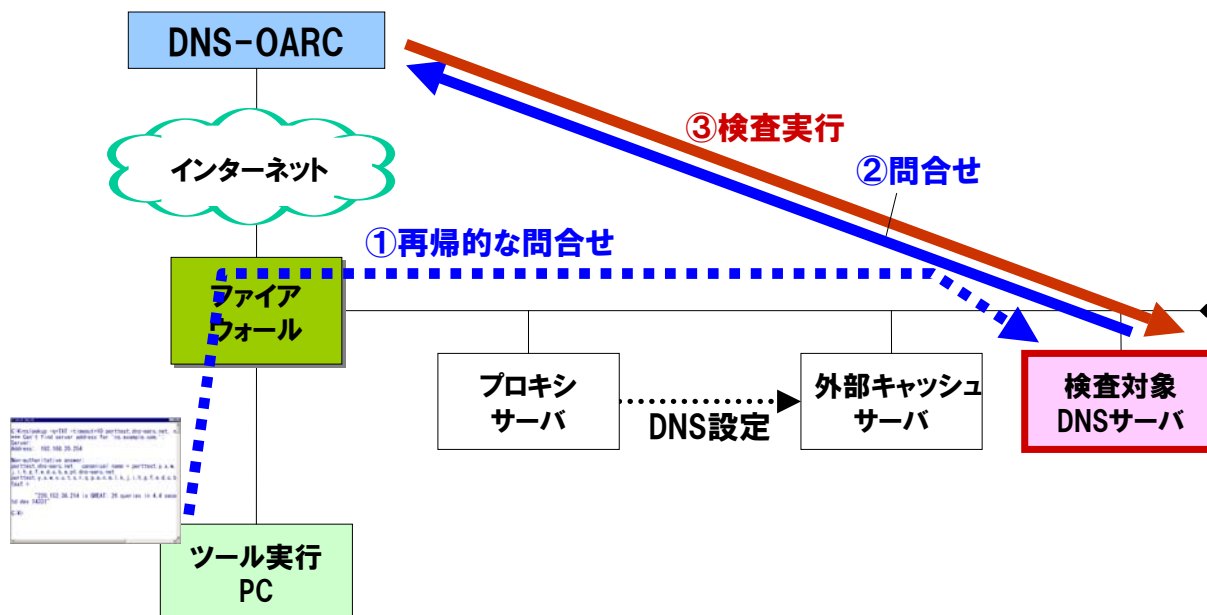
```
"192.168.0.2 is GREAT: 26 queries in 3.6 seconds from 26 ports  
with std dev 17332"
```

```
y.x.w.v.u.t.s.r.q.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.pt.dns-oarc.net  
nameserver = ns.y.x.w.v.u.t.s.r.q.p.o.n.m.l.k.j.i.h.g.f.e.d.c.b.a.pt.dns-oarc.net
```

「DNS-OARC Randomness Test(コマンドライン版)」の実行例です。「ns.example.com」の送信元ポート番号のランダム性を検査した結果、「GREAT」であることがわかります。

### 3.3 DNS-OARC(コマンドライン版)の注意点(1/2)

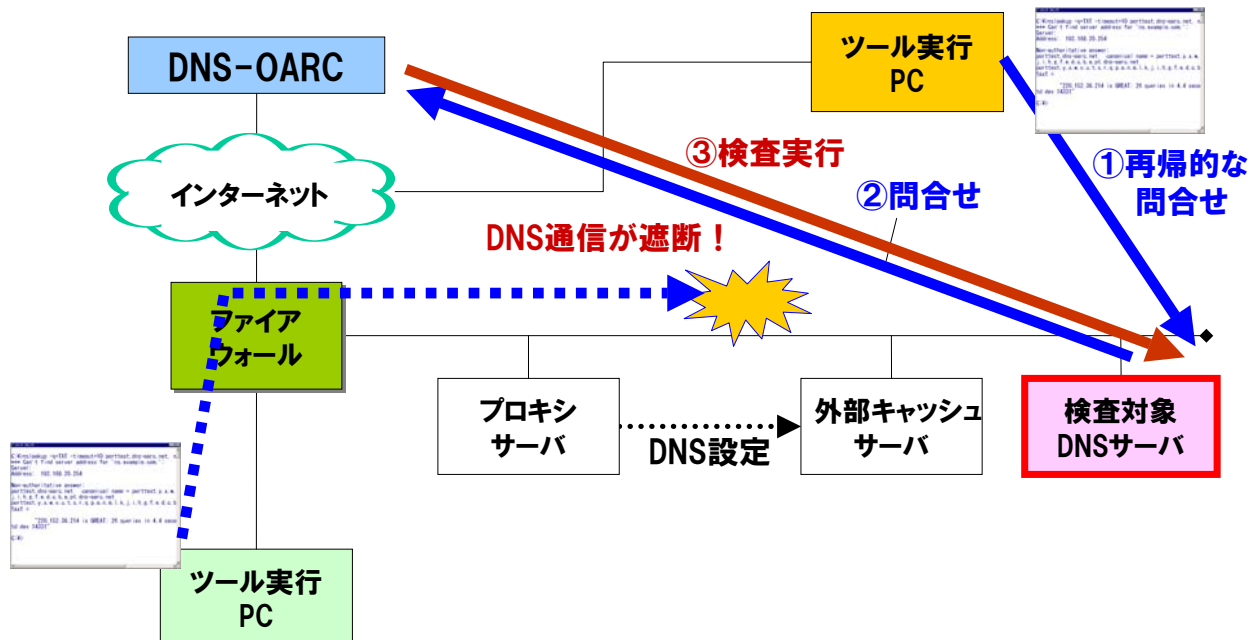
コマンドライン版では、Web版(プロキシ経由)では検査対象とならないキャッシュサーバをイントラネット内から検査可能です。ただし、**ツール実行PCと検査対象DNSサーバ間のDNS通信が遮断されていない場合に限り**ます。



「DNS-OARC Randomness Test(コマンドライン版)」使用時の注意点です。検査できるのは、ツール実行PCと検査対象DNSサーバ間のDNS通信が遮断されていない場合に限り

### 3.3 DNS-OARC(コマンドライン版)の注意点(2/2)

ファイアウォールなどにより、ツール実行PCと検査対象DNSサーバ間のDNS通信が遮断されている場合は、イントラネットから正しく検査が行えません。その場合は、インターネットに直接接続しているPCから実行してください。



ツール実行PCと検査対象DNSサーバ間のDNS通信が遮断されている場合は検査できませんので、インターネットに直接接続しているツール実行PCから検査する必要があります。

### 検査ツールでの確認は終了しましたか？

- ◇ Cross-Pollination Checkツールでは、コンテンツサーバに対して、1)再帰動作の可否、2)送信元ポート番号のランダム性有無を検査できます。
- ◇ DNS-OARCツールでは、キャッシュサーバに対して、1)送信元ポート番号のランダム性有無、2)トランザクションIDのランダム性有無を検査できます。
- ◇ コンテンツサーバの検査には、Cross-Pollination Checkを使用しましょう。
- ◇ キャッシュサーバの検査には、DNS-OARC Randomness Test (Web版)を使用しましょう。
- ◇ Cross-Pollination CheckとDNS-OARC Randomness Test (Web版)のいずれでも検査できない場合は、DNS-OARC Randomness Test (コマンドライン版)を使用しましょう。  
その際、環境によっては、イントラネットからは正しく検査ができない場合があります、インターネットに直接接続しているPCから検査を行う必要があります。

DNSキャッシュポイズニング対策の検査ツールに関するポイントをまとめてありますので、ぜひ、検査ツールで確認してみてください。

## 4. 再帰動作の設定

### 4.1 BIND DNSサーバでの対策

### 4.2 Windows DNSサーバでの対策

### 4.3 まとめ

「4. 再帰動作の設定」では、BINDとWindows DNSサーバの再帰動作の設定について説明します。

事前に、次に示す用語の意味を理解しておく必要があります。

- コンテンツサーバ
- キャッシュサーバ
- 再帰的な問合せ

BIND DNSサーバでの対策ポイントは次の通りです。

### コンテンツサーバ

再帰動作が無効になっていることを確認する。

### キャッシュ兼コンテンツサーバ

コンテンツサーバ単独(再帰動作を無効とし、キャッシュサーバとして動作させない、あるいは、キャッシュサーバとコンテンツサーバを物理装置的に分離する)でのサーバ稼働を検討する。コンテンツサーバ単独での稼働が可能な場合には、「コンテンツサーバでの対策」を実施する。

キャッシュサーバ兼コンテンツサーバで運用する必要がある場合には、再帰的な問合せは、イントラネットからのアクセスのみを許可する。

### キャッシュサーバ

再帰的な問合せは、イントラネットからのアクセスのみを許可する。

BIND DNSサーバでの対策では、次の資料を基に、コンテンツサーバ、キャッシュ兼コンテンツサーバ、キャッシュサーバの3つにわけ、再帰的な問合せに関するアクセス制御設定について説明します。

- DNSの再帰的な問合せを使ったDDoS攻撃の対策について

<http://jprs.jp/tech/notice/2006-03-29-dns-cache-server.html>

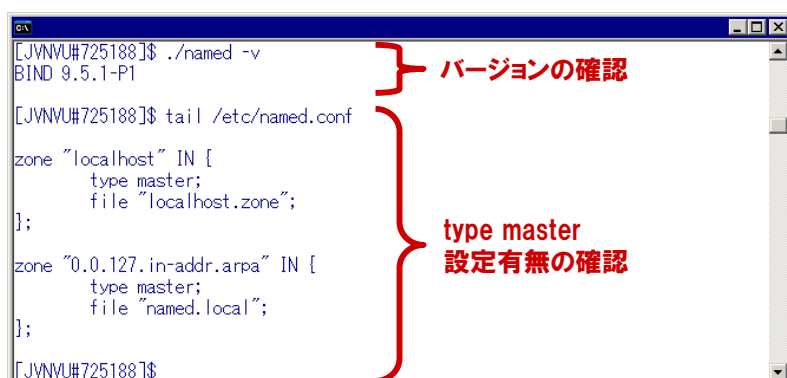
対策にあたっては、製品開発ベンダから提供されているJVNVU#800113対応のセキュリティ修正プログラムの適用を前提とします。なお、ISC (Internet Systems Consortium) BIND 9については、サービス運用妨害 (DoS) の脆弱性 (JVNVU#725188) が報告されています。named.confに type master; の設定がある場合には脆弱性の影響を受けることとなりますのでJVNVU#725188対応のセキュリティ修正プログラムの適用も検討してください。

- 複数のDNS実装にキャッシュポイズニングの脆弱性 (公開日: 2008/07/09)

<http://jvn.jp/cert/JVNVU800113/>

- ISC BIND 9におけるサービス運用妨害 (DoS) の脆弱性 (公開日: 2009/07/29)

<http://jvn.jp/cert/JVNVU725188/>



```
[JVNVU#725188]$ ./named -v
BIND 9.5.1-P1

[JVNVU#725188]$ tail /etc/named.conf

zone "localhost" IN {
    type master;
    file "localhost.zone";
};

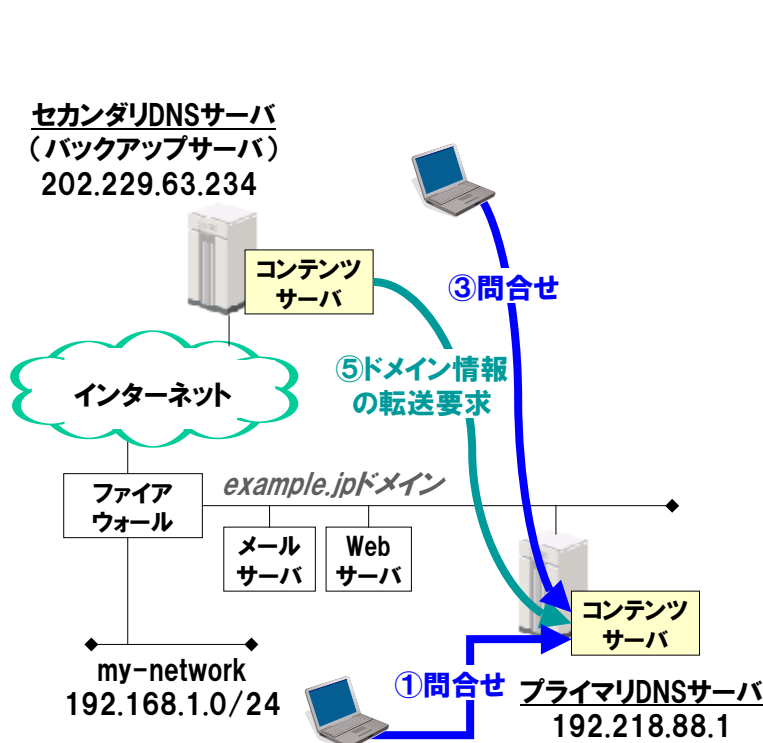
zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "named.local";
};

[JVNVU#725188]$
```

バージョンの確認

type master  
設定有無の確認

## 4.1 コンテンツサーバでの対策



```
// グローバルオプションの設定
options {
    fetch-glue no ; // BIND 9 では不要
    recursion no ;
    directory "/etc/ns" ;
    allow-transfer { none ; } ;
};
// example.jp のマスタ DNS サーバ設定
zone "example.jp" {
    type master ;
    file "example.jp.zone" ;
    allow-transfer { 202.229.63.234 ; } ;
};
// ルートサーバへの hint 情報
zone "." {
    type hint ;
    file "/dev/null" ;
    // ファイル名に /dev/null を指定
};
```

コンテンツサーバでは、イントラネットからの問合せ (①)、インターネットからの問合せ (③)、セカンダリDNSサーバからのドメイン情報の転送(ゾーン転送)要求 (⑤) を許可します。

### 【 設定ファイルのポイント 】

- recursion no ;

再帰動作を無効とすることで、問合せ (①③) のみを受け付けるよう設定します。この設定の場合、問合せに対してアクセス制限を行う必要はありません。

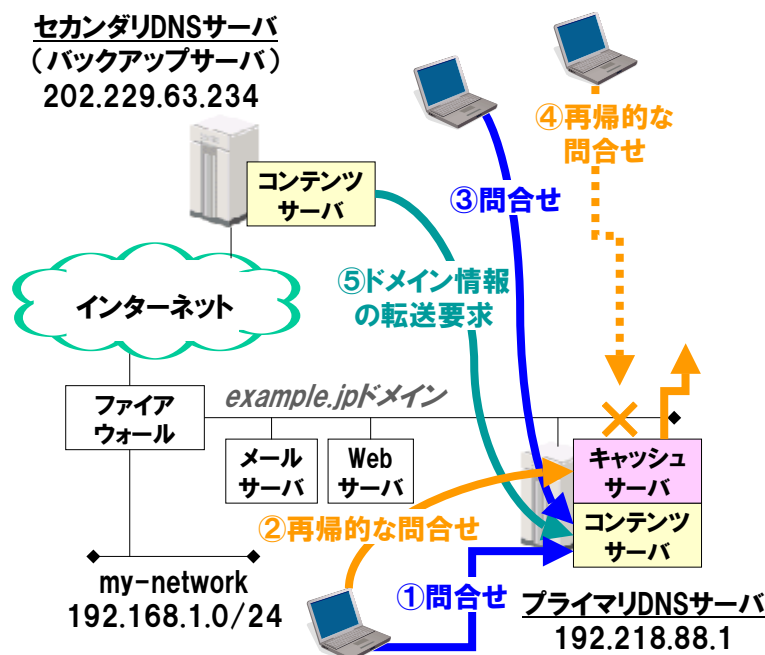
- allow-transfer { 202.229.63.234 ; } ;

ドメイン情報の転送(ゾーン転送) (⑤) をセカンダリDNSサーバのみに制限します。

- file "/dev/null" ;

ルートサーバへの hint 情報は、再帰動作を有効としている場合のみ必要になります。ここでは、利用しないことを明示するために、ファイル名として /dev/null を指定します。

## 4.1 キャッシュ兼コンテンツサーバでの対策



```
// イン트라ネットからのアクセス設定
acl my-network {
    192.168.1.0/24 ;
};
// グローバルオプションの設定
options {
    fetch-glue no ; // BIND 9では不要
    recursion yes ;
    directory "/etc/ns" ;
    allow-query {
        localhost ;
        my-network ;
    };
    allow-transfer { none ; };
};
// example.jp のプライマリ DNS サーバ設定
zone "example.jp" {
    type master ;
    file "example.jp.zone" ;
    allow-query { any ; };
    allow-transfer { 202.229.63.234 ; };
};
zone "." {
    type hint ;
    file "named.root" ;
};
```

再帰動作を期待する問合せを「再帰的な問合せ」と記載しています。

キャッシュ兼コンテンツサーバでは、イントラネットからの問合せ(①)と再帰的な問合せ(②)、インターネットからの問合せ(③)、セカンダリDNSサーバからのドメイン情報の転送(ゾーン転送)要求(⑤)を許可します。また、インターネットからの再帰的な問合せ(④)を拒否します。

### 【 設定ファイルのポイント 】

- `acl my-network { 192.168.1.0/24 ; } ;`

イントラネットを定義します。

- `recursion yes ;`

キャッシュサーバとして稼働させるため、再帰動作を有効とします。

- `allow-query { localhost ; my-network ; } ;`

`allow-query`を用いて問合せ(含む、再帰的な問合せ)のアクセス制御を実施します。キャッシュ兼コンテンツサーバ(`localhost`)とイントラネット(`my-network`)からの問合せ(含む、再帰的な問合せ)(①②)のみを許可します。

- `allow-query { any ; } ;`

`example.jp`ドメインに関する問合せのみ、イントラネットからの問合せ(①)とインターネットからの問合せ(③)のいずれも許可します。

- `allow-transfer { 202.229.63.234 ; } ;`

ドメイン情報の転送(ゾーン転送)(⑤)をセカンダリDNSサーバのみに制限します。

- `file "named.root" ;`

再帰動作を有効としているので、ルートサーバへの `hint` 情報が格納されたファイルを指定します。最新の `hint` 情報は、次のURLから取得してください。

`ftp://ftp.internic.net/domain/named.root`

## 4.1 キャッシュ兼コンテンツサーバでの対策（留意事項1）



BIND9.2.6ならびに、それ以前の実装で `allow-query` を使ってアクセス制御した場合、アクセス制御は有効に機能するのですが(`status: REFUSED`)、再帰動作の有効フラグ(`ra: Recursion available`)がON(`flags: ra`)となるため、Cross-Pollination Check(<http://recursive.iana.org/>)では、“Vulnerable(Is recursive, could not detect source port randomisation)”と判定されます。

```
>>> DiG <<<> @bind926.ipa.go.jp. www.example.com.
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 62833
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
```

BIND最新版では、この問題は解決されていますので、BIND最新版にアップデートすることを推奨します。なお、`allow-recursion`を併記することで再帰動作の有効フラグをOFFできますが、あくまでも、暫定対策として利用してください。

```
options {
    fetch-glue no ; // BIND 9では不要
    recursion yes ;
    directory "/etc/ns" ;
    allow-query { localhost ; my-network ; } ;
    allow-recursion { localhost ; my-network ; } ; // BIND 9.2.6以前への暫定対策
    allow-transfer { none ; } ;
};
```

古いバージョンのBINDを利用している場合の留意事項です。

BIND9.2.6ならびに、それ以前の実装で `allow-query` を使ってアクセス制御した場合、アクセス制御は有効に機能するのですが(`status: REFUSED`)、回答に格納される再帰動作の有効フラグ(`ra: Recursion available`)がON(`flags: ra`)となるため、Cross-Pollination Check(<http://recursive.iana.org/>)では、“Vulnerable(Is recursive, could not detect source port randomisation)”と判定されます。

なお、フラグの情報については、`dig`コマンドを使って確認できます。

使い方:        `dig @ [使用するDNSサーバ] [検索するホスト名]`  
                 `dig @bind926.ipa.go.jp. www.example.com.`

BIND最新版では、この問題は解決されていますので、BIND最新版にアップデートすることを推奨します。なお、`allow-recursion`を併記することで再帰動作の有効フラグをOFFできますが、あくまでも、暫定対策として利用してください。

- `allow-query { localhost ; my-network ; } ;`

`allow-query`を用いて問合せ(含む、再帰的な問合せ)のアクセス制御を実施します。キャッシュ兼コンテンツサーバ(ローカルホスト)とイントラネットからの問合せ(含む、再帰的な問合せ) (①②)のみを許可します。

- `allow-recursion { localhost ; my-network ; } ;`

`allow-recursion`を用いて再帰的な問合せに対してアクセス制御を実施します。キャッシュ兼コンテンツサーバ(ローカルホスト)とイントラネットからの再帰的な問合せ (②)のみを許可します。

### allow-queryとallow-recursionのアクセス制御の違い

allow-queryを使用したアクセス制御の場合には、再帰的な問合せ自身を拒否 (status: REFUSED) し、何もデータを含まない回答を返信します。

```
; <<>> DiG <<>> @allow-query.ipa.go.jp. www.example.com.
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 54392
;; flags: qr rd: QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
```

allow-recursionを使用したアクセス制御の場合には、再帰的な問合せを受入れます (status: NOERROR)。ただし、名前解決をせず (ANSWER: 0)、次に問合せるべきDNSサーバ (AUTHORITY: 2, ADDITIONAL: 2) を返信します。

```
; <<>> DiG <<>> @allow-recursion.ipa.go.jp. www.example.com.
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 535
;; flags: qr rd: QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2
```

digコマンドの結果を使って、allow-queryとallow-recursionのアクセス制御の違いについて説明します。

#### 【 アクセス制御なしの場合 】

アクセス制御なしの場合には、再帰的な問合せを受入れ (status: NOERROR)、名前解決 (ANSWER: 1) を実施します。

```
; <<>> DiG <<>> @any.ipa.go.jp www.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7322
;; flags: qr rd ra: QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
```

#### 【 allow-queryの場合 】

allow-queryを使用したアクセス制御の場合には、許可されていないネットワークからの再帰的な問合せに対して、再帰的な問合せ自身を拒否 (status: REFUSED) し、何もデータを含まない回答 (ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0) を返信します。

#### 【 allow-recursionの場合 】

allow-recursionを使用したアクセス制御の場合には、許可されていないネットワークからの再帰的な問合せに対して、再帰的な問合せを受入れます (status: NOERROR)。ただし、名前解決をせず (ANSWER: 0)、名前解決処理を進めるために、次に問合せるべきDNSサーバ (AUTHORITY: 2, ADDITIONAL: 2) を返信します。

## 4.1 キャッシュ兼コンテンツサーバでの対策

**allow-queryを用いたアクセス制御を推奨します。**

```
コマンド プロンプト
C:\>nslookup -q=NS . allow-recursion.ipa.go.jp.
Server: allow-recursion.ipa.go.jp.
Address: 10.10.10.17

Non-authoritative answer:
(root) nameserver = B.ROOT-SERVERS.NET
(root) nameserver = C.ROOT-SERVERS.NET
(root) nameserver = D.ROOT-SERVERS.NET
(root) nameserver = E.ROOT-SERVERS.NET
(root) nameserver = F.ROOT-SERVERS.NET
(root) nameserver = G.ROOT-SERVERS.NET
(root) nameserver = H.ROOT-SERVERS.NET
(root) nameserver = I.ROOT-SERVERS.NET
(root) nameserver = J.ROOT-SERVERS.NET
(root) nameserver = K.ROOT-SERVERS.NET
(root) nameserver = L.ROOT-SERVERS.NET
(root) nameserver = M.ROOT-SERVERS.NET
(root) nameserver = A.ROOT-SERVERS.NET

L.ROOT-SERVERS.NET internet address = 199.7.83.42
L.ROOT-SERVERS.NET AAAA IPv6 address = 2001:500:3::42
M.ROOT-SERVERS.NET internet address = 202.12.27.33
M.ROOT-SERVERS.NET AAAA IPv6 address = 2001:dc3::35

C:\>nslookup -q=NS . allow-query.ipa.go.jp.
Server: allow-query.ipa.go.jp.
Address: 10.10.10.18

*** allow-query.ipa.go.jp can't find .: Query refused
```

**allow-recursionを用いた  
アクセス制御の場合、  
再帰的な問合せを受け入れて  
しまいます。**

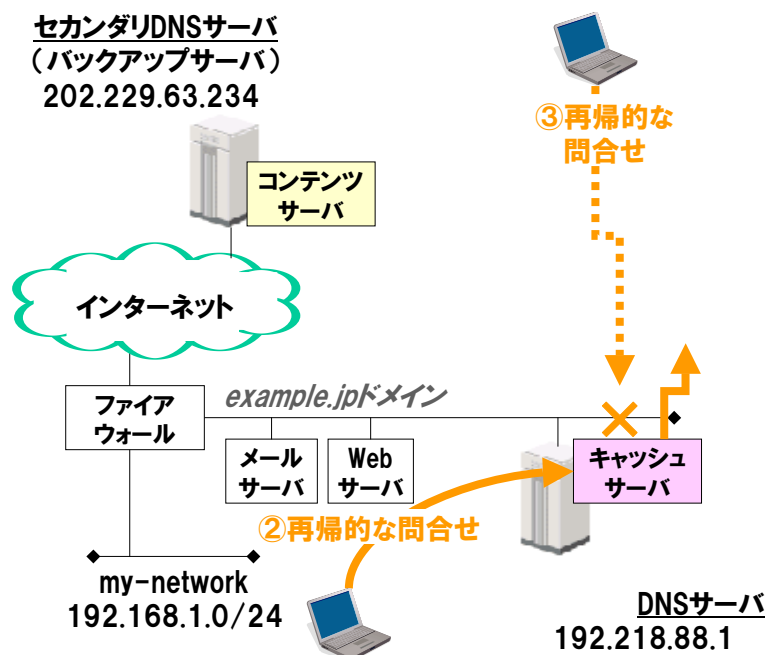
**allow-queryを用いた  
アクセス制御の場合、  
再帰的な問合せを拒否できます。**

2009年にはいつから、DDoS (Distributed DoS) 攻撃に活用するため、外部からの再帰的な問合せに回答してしまうキャッシュサーバに対して、ルートサーバ(名前解決の基点となるDNSサーバ)のIPアドレスを繰り返し要求する活動が報告されています。BINDを用いて構築したキャッシュサーバでは、allow-recursionを用いたアクセス制御ではなく、問合せ(含む、再帰的な問合せ)自身を拒否するallow-queryを用いたアクセス制御を推奨します。

報告された活動については、次の資料を参照してください。

- DNS queries for "."  
<http://isc.sans.org/diary.html?storyid=5713>
- DNS DDoS - let's use a long term solution  
<http://isc.sans.org/diary.html?storyid=5773>

## 4.1 キャッシュサーバでの対策



```
// イン트라ネットからのアクセス設定
acl my-network {
    192.168.1.0/24 ;
};
// グローバルオプションの設定
options {
    fetch-glue no ; // BIND 9では不要
    recursion yes ;
    directory "/etc/ns" ;
    allow-query {
        localhost ;
        my-network ;
    } ;
};
// ルートサーバへの hint 情報
zone "." {
    type hint ;
    file "named.root" ;
};
```

再帰動作を期待する問合せを「再帰的な問合せ」と記載しています。

キャッシュサーバでは、イントラネットからの再帰的な問合せ(②)を許可します。また、インターネットからの再帰的な問合せ(④)を拒否します。

### 【 設定ファイルのポイント 】

- `acl my-network { 192.168.1.0/24 ; } ;`

イントラネットを定義します。

- `recursion yes ;`

キャッシュサーバとして稼働させるため、再帰動作を有効とします。

- `allow-query { localhost ; my-network ; } ;`

`allow-query`を用いて問合せ(含む、再帰的な問合せ)のアクセス制御を実施します。キャッシュサーバ(`localhost`)とイントラネット(`my-network`)からの問合せ(含む、再帰的な問合せ)(②)のみを許可します。

- `file "named.root" ;`

再帰動作を有効としているので、ルートサーバへの hint 情報が格納されたファイルを指定します。最新の hint 情報は、次のURLから取得してください。

`ftp://ftp.internic.net/domain/named.root`

Windows DNSサーバでの対策ポイントは次の通りです。

### コンテンツサーバ

再帰動作が無効になっていることを確認する。

### キャッシュ兼コンテンツサーバ

Windows DNSサーバは、再帰的な問合せを受け付けるか／否かしか設定できないため(BIND DNSサーバのような細かなアクセス制御機能なし)、キャッシュサーバとコンテンツサーバを物理装置的に分離して運用する。

### キャッシュサーバ

ファイアウォールなどのパケットフィルタリング機能を用いて、イントラネットからの再帰的な問合せのみを許可するよう制限する。

Windows DNSサーバでの対策では、次の資料を基に、コンテンツサーバ、キャッシュ兼コンテンツサーバ、キャッシュサーバの3つにわけ、再帰動作に関するアクセス制御設定について説明します。

- DNS > DNSの操作方法

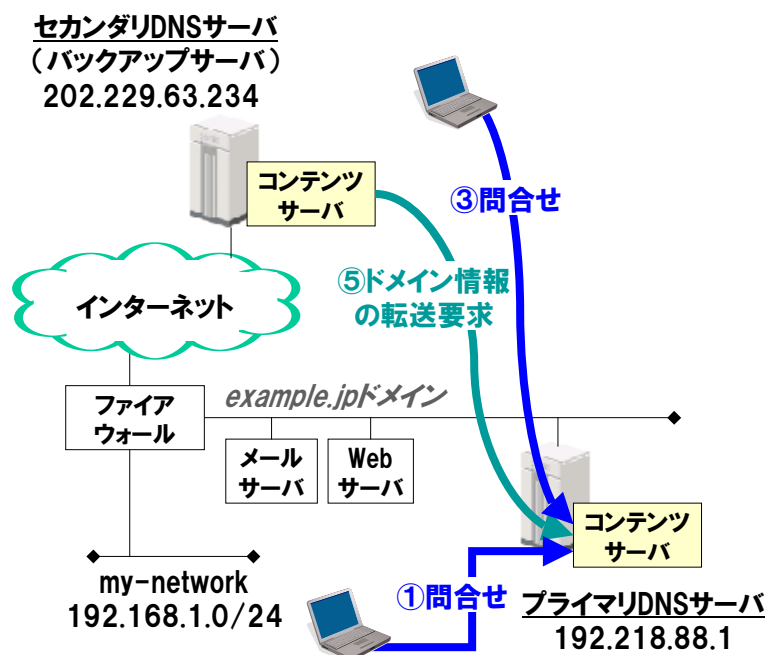
<http://technet.microsoft.com/ja-jp/library/cc757540.aspx>

なお、対策にあたっては、製品開発ベンダから提供されているセキュリティ修正プログラムの適用を前提とします。

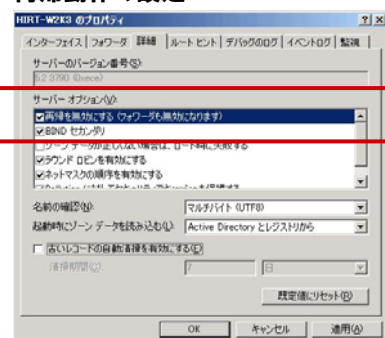
- 複数のDNS実装にキャッシュポイズニングの脆弱性

<http://jvn.jp/cert/JVNVU800113/>

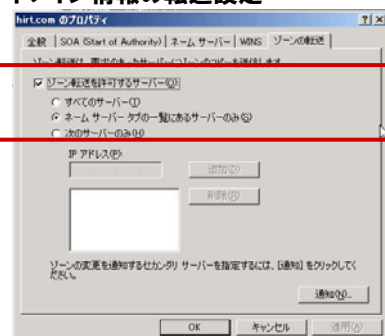
## 4.2 コンテンツサーバでの対策



### // 再帰動作の設定



### // ドメイン情報の転送設定



コンテンツサーバでは、イントラネットからの問合せ(①)、インターネットからの問合せ(③)、セカンダリDNSサーバからのドメイン情報の転送(ゾーン転送)要求(⑤)を許可します。

### 【 設定のポイント 】

#### ● 再帰動作の設定

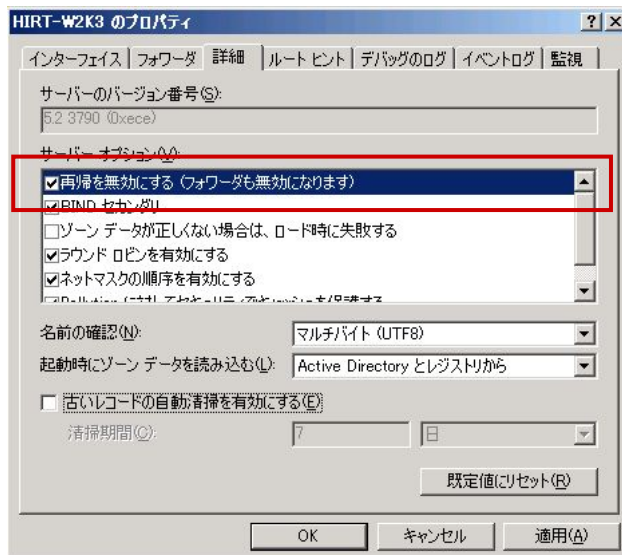
「再帰を無効にする(フォワーダも無効になります)」のチェックボックスをONにします。

#### ● ドメイン情報の転送設定

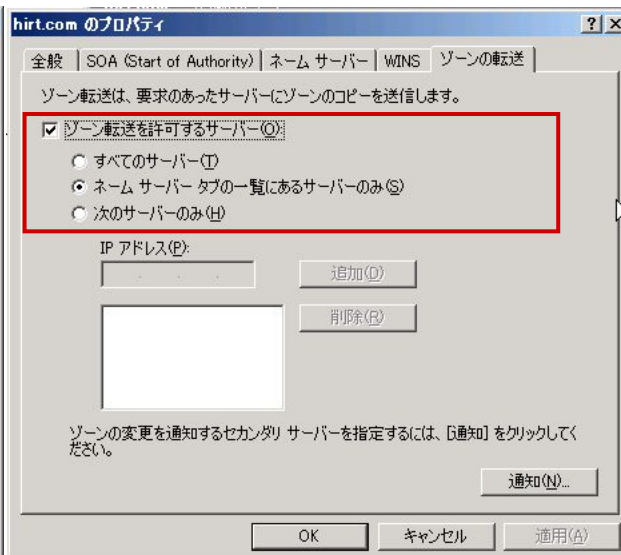
ゾーン転送を許可するサーバーのチェックボックスをONにします。また、「ネームサーバタブの一覧にあるサーバーのみ」「次のサーバーのみ」のいずれかを使用して、ドメイン情報の転送(ゾーン転送)(⑤)をセカンダリDNSサーバのみに制限します。

## 4.2 コンテンツサーバでの対策(設定画面拡大)

### // 再帰動作の設定



### // ドメイン情報の転送設定



### 【 設定のポイント 】

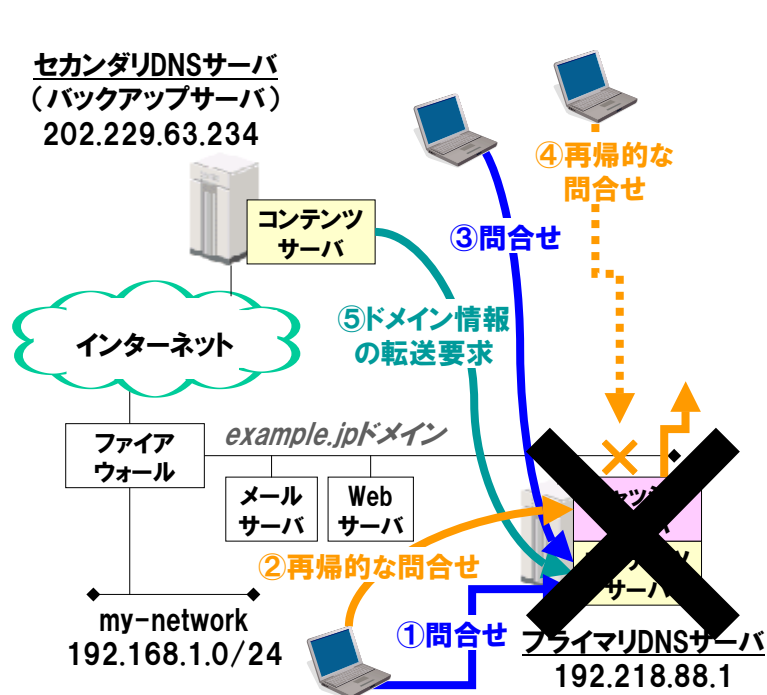
#### ● 再帰動作の設定

「再帰を無効にする(フォワーダも無効になります)」のチェックボックスをONにします。

#### ● ドメイン情報の転送設定

ゾーン転送を許可するサーバのチェックボックスをONにします。また、「ネームサーバタブの一覧にあるサーバーのみ」「次のサーバーのみ」のいずれかを使用して、ドメイン情報の転送(ゾーン転送) (⑤) をセカンダリDNSサーバのみに制限します。

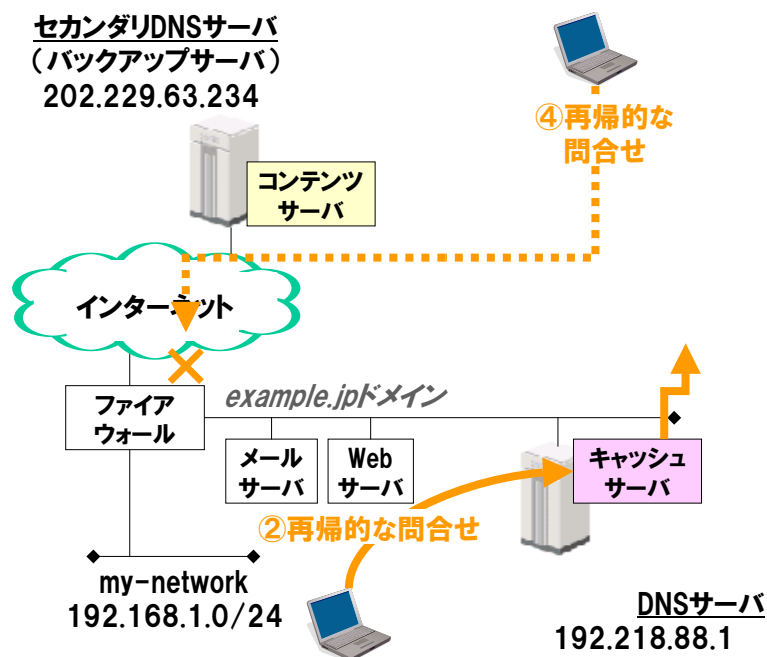
## 4.2 キャッシュ兼コンテンツサーバでの対策



Windows DNSサーバは、再帰的な問合せを受け付けるか／否かしか設定できないため(BIND DNSサーバのような細かなアクセス制御機能なし)、キャッシュサーバとコンテンツサーバを物理装置的に分離して運用します。

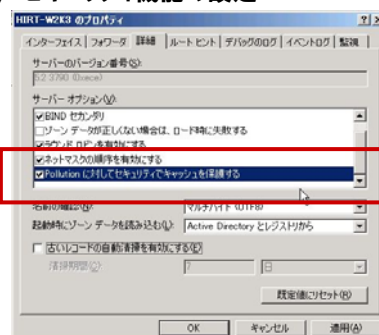
Windows DNSサーバの場合には、キャッシュサーバとコンテンツサーバを物理装置的に分離した運用を推奨します。

## 4.2 キャッシュサーバでの対策



ファイアウォール製品などのパケットフィルタリング機能を用いて、イントラネットからの再帰的な問合せのみを許可するよう制限します。また、Windows DNSサーバのセキュリティ機能を活用します。

### // セキュリティ機能の設定



再帰動作を期待する問合せを「再帰的な問合せ」と記載しています。

キャッシュサーバでは、イントラネットからの再帰的な問合せ(②)を許可します。また、インターネットからの再帰的な問合せ(④)を拒否します。

### 【 設定のポイント 】

#### ● 再帰動作の設定

「再帰を無効にする(フォワーダも無効になります)」のチェックボックスをOFFにします。

#### ● ファイアウォールなどを用いたアクセス制御の実施

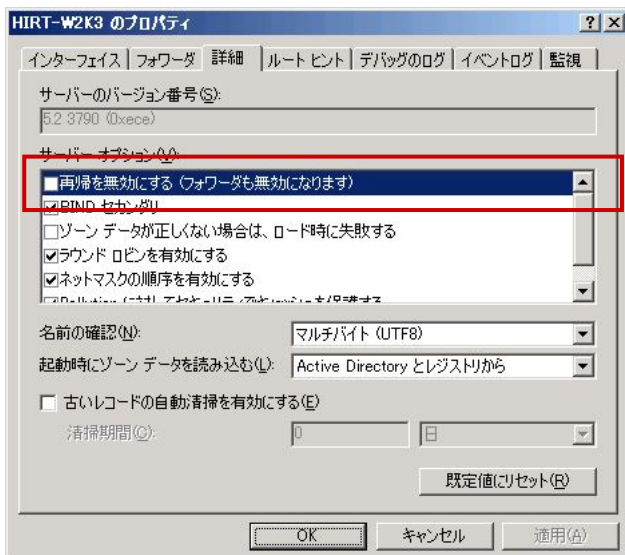
Windows DNSサーバは、再帰的な問合せを受け付けるか／否かしか設定できません。イントラネットからの再帰的な問合せ(②)を許可し、インターネットからの再帰的な問合せ(④)を拒否するために、ファイアウォールなどを用いて、アクセス制御を実施してください。

#### ● セキュリティ機能の設定

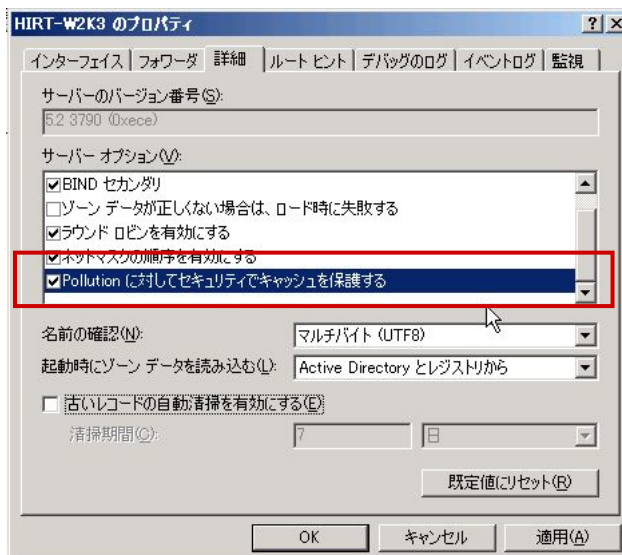
「Pollution に対してセキュリティでキャッシュを保護する」のチェックボックスをONにします。この機能は、回答に含まれている無関係なデータ(コンテンツサーバが管理する原本以外のデータ)をキャッシュしないという機能です。

## 4.2 キャッシュサーバでの対策(設定画面拡大)

### // 再帰動作の設定



### // セキュリティ機能の設定



### 【 設定のポイント 】

#### 再帰動作の設定

「再帰を無効にする(フォワーダも無効になります)」のチェックボックスをOFFにします。

#### セキュリティ機能の設定

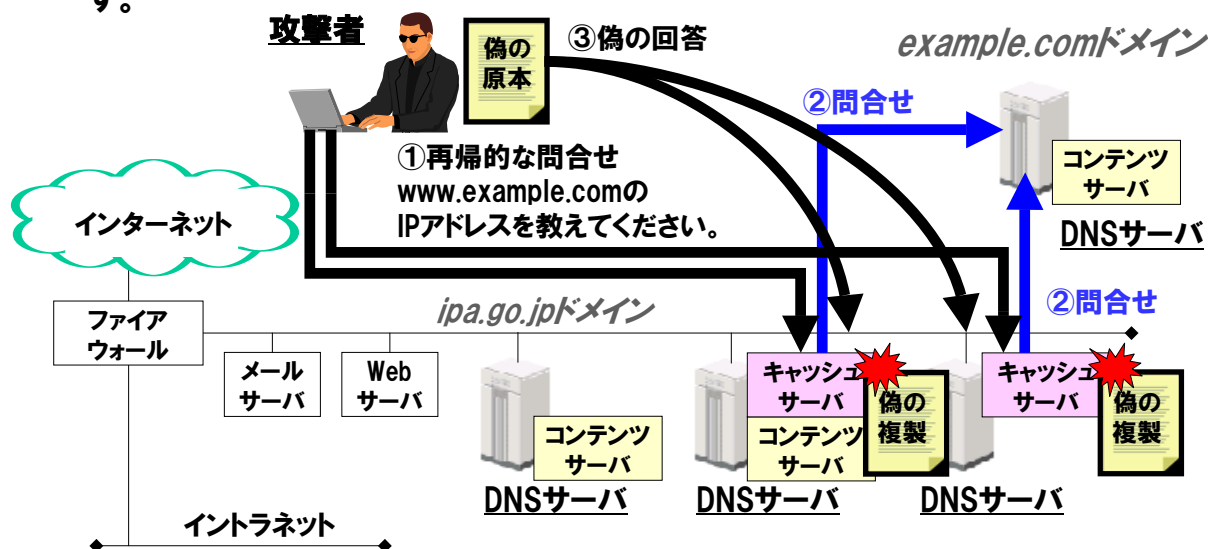
セキュリティ機能の設定詳細については、次の資料を参照してください。

- サーバーのキャッシュを名前の汚染から保護する  
<http://technet.microsoft.com/ja-jp/library/cc758810.aspx>
- DNS サーバーの [Pollution に対してセキュリティでキャッシュを保護する] 設定について  
<http://support.microsoft.com/kb/316786/ja>

「Pollution に対してセキュリティでキャッシュを保護する」は、回答に含まれている無関係なデータ(コンテンツサーバが管理する原本以外のデータ)をキャッシュしないという機能です。例えば、ドメインexample.comを管理するコンテンツサーバからの回答の場合には、example.com以外のデータ(www.ipa.go.jpなど)はキャッシュしないこととなります。

## DNSサーバの設定は適切ですか？

キャッシュサーバとして動作しているDNSサーバが、インターネットからの再帰的な問合せ(①)に対して再帰動作による問合せ(②)をしてしまう場合、DNSキャッシュポイズニング攻撃(③)を受ける可能性が高くなります。



DNSサーバのアクセス制御設定についてまとめます。

キャッシュサーバとして動作しているDNSサーバが、インターネットからの再帰的な問合せ(①)に対して再帰動作による問合せ(②)をしてしまう場合、DNSキャッシュポイズニング攻撃を受ける可能性が高くなります。基本的なアクセス制御設定は、インターネットからの再帰的な問合せ(①)に対する再帰動作を無効にすることです。

DNSサーバの設定を再確認して、安全なDNSサーバによる基盤サービスを実現していきましょう。

- [発行] 2009年1月14日 第1版:新規  
 2009年2月6日 第2版:DNS-OARCの使い方(P26)とDDoS対策の注意事項(P40)を追記  
 2009年8月11日 第3版:RFC1035にあわせた用語定義(P3 他)に変更  
 JNVU#725188(P35)を追記  
 IPA(独立行政法人情報処理推進機構)セキュリティセンター  
 [執筆] 寺田 真敏

