DNSのセキュリティ

DNSの仕様に起因するセキュリティ問題

（株）インターネットイニシアティブ
其田　学

本セクションのフォーカス

キャッシュポイズニング（毒入れ）をどうやって防ぐか

キャッシュポイズニングとは

- キャッシュDNSサーバに対して不正なレコードをキャッシュさせ
- クライアントに不正な応答を返してしまう事。

どうすれば、キャッシュポイズニングを防ぐことができるのか

結論

本日の結論

- DNSプロトコルは完全性の保証が無いので、毒入れを完全に防ぐことは不可能。
- 対策を施す事により毒入れの成功確率を下げ、リスクを下げる事は可能
- どうしても守りたいドメイン名に対しては、DNSSECを導入する。

アジェンダ

- DNSプロトコルの復習
- 主な毒入れ手法
- 対策
- まとめ

DNSプロトコルの復習

DNSプロトコルの復習

- セキュリティ的な性質
- 委任
- 問い合わせの種類
- 問い合わせの複雑性について
- 権威とRanking Data

セキュリティ的な性質

- 機密性
  - 無し、DNSは公開データ
- 完全性
  - 保証無し、内容の検証ができない。
- 可用性
  - 非常に高い可用性が求められる。

完全性の保証が無い為、毒入れが可能

委任

- 親のゾーンから一部のゾーンを分ける事をゾーン分割(zone cut)といい、分割したゾーンを他の権威DNSサーバに管理を任せることを委任といいます。

委任

- この委任情報の部分のやり取りはDNSのパケットで行われてます。
- 委任元は、委任先の権威DNSサーバの情報として、NSレコードとして、応答します。

委任

- 委任したホスト名が委任したゾーンの内部名の場合は、通常はglue（糊付け）レコードしてアドレスレコードをゾーンに持っておりこれも応答します。

例

```
; <<>> DiG 9.10.0 <<>> +norec +noedns +nodnssec @d.dns.jp iij.ad.jp
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34458
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;iij.ad.jp.                     IN      A

;; AUTHORITY SECTION:
iij.ad.jp.              86400 IN      NS      dns1.iij.ad.jp.
iij.ad.jp.              86400 IN      NS      dns0.iij.ad.jp.

;; ADDITIONAL SECTION:
dns0.iij.ad.jp.         86400 IN      A       210.130.0.5
dns1.iij.ad.jp.         86400 IN      A       210.130.1.5
dns0.iij.ad.jp.         86400 IN      AAAA    2001:240::105
dns1.iij.ad.jp.         86400 IN      AAAA    2001:240::115
```

iij.ad.jp.をdns1.iij.ad.jp.とdns0.iij.ad.jp.に委任している。
dns1.iij.ad.jp.とdns0.iij.ad.jp.はiij.ad.jpの内部名なので応答にglueレコードが入る。

委任

- 委任した権威DNSサーバのホスト名が委任したゾーンの外部名の場合は、アドレスレコードが出るかはケース次第

例 アドレスレコードが付くパターン

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 58748
;; flags: qr ; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 4
;; QUESTION SECTION:
;; iijmio.jp.  IN      A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
iijmio.jp.      86400       IN      NS      dns0.iij.ad.jp.
iijmio.jp.      86400       IN      NS      dns1.iij.ad.jp.

;; ADDITIONAL SECTION:
dns0.iij.ad.jp.     86400       IN      A       210.130.0.5
dns1.iij.ad.jp.     86400       IN      A       210.130.1.5
dns0.iij.ad.jp.     86400       IN      AAAA    2001:240::105
dns1.iij.ad.jp.     86400       IN      AAAA    2001:240::115
```

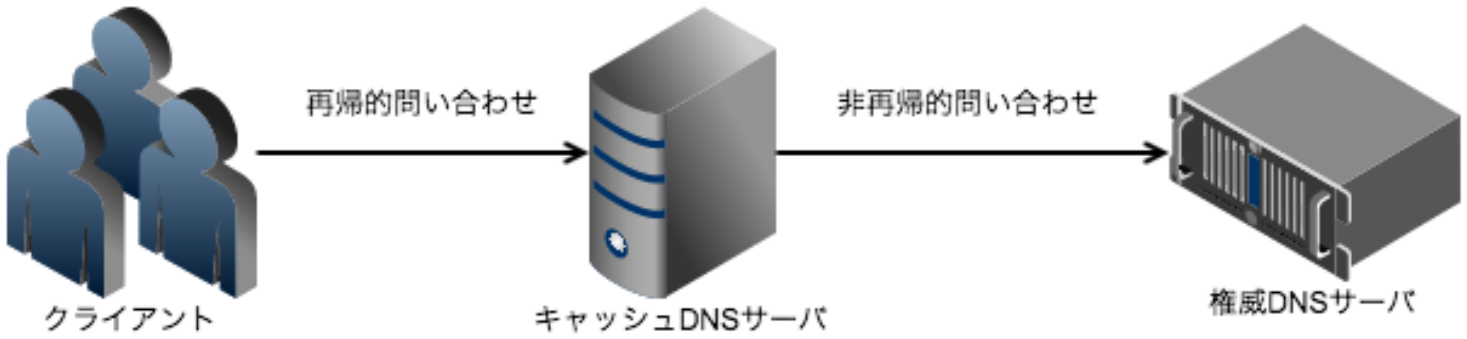
jpゾーンに、iij.ad.jp用のglueとして、dns[01].iij.ad.jpがあるため、
外部名では有るがNSのアドレスレコードの応答を返す。

例 アドレスレコードが無いパターン

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 13092
;; flags: qr ; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;; iij.jp.  IN   A

;; ANSWER SECTION:

;; AUTHORITY SECTION:
iij.jp.     86400   IN   NS  dns-b.iij.ad.jp.
iij.jp.     86400   IN   NS  dns-c.iij.ad.jp.
```

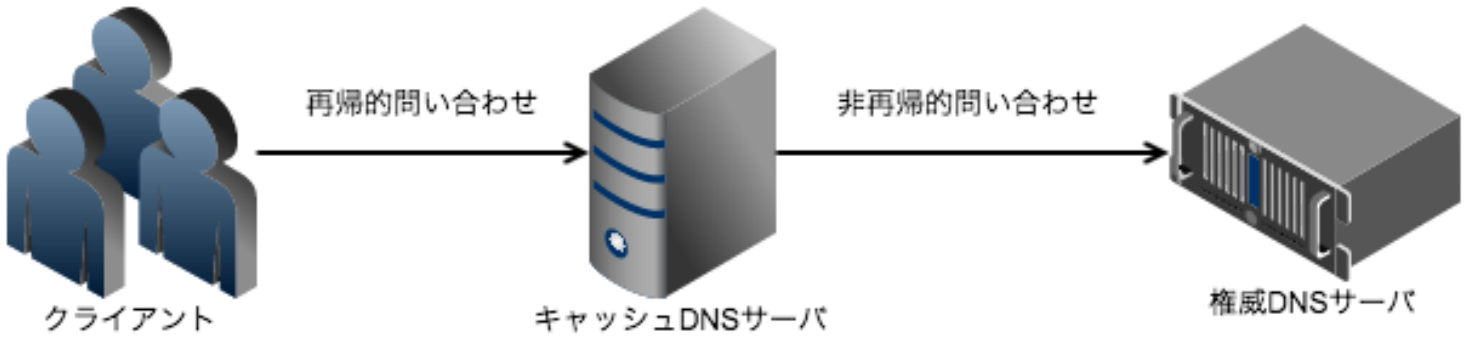
問い合わせの種類について

- DNSの問い合わせ（クエリー）は2種類あります。
  - 再帰的問い合わせ
  - 非再帰的問い合わせ



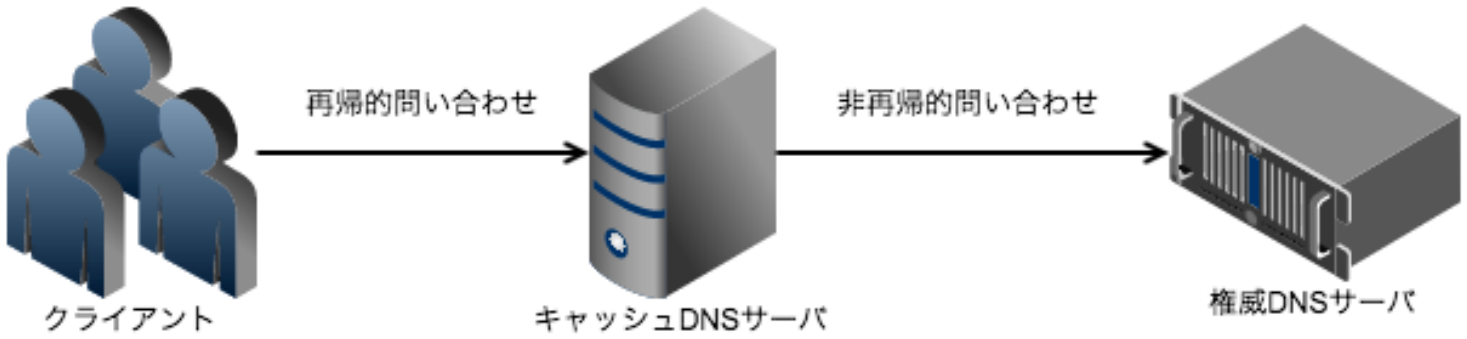
問い合わせの種類について

- 再帰的問い合わせはクライアントがキャッシュDNSサーバに対して、名前解決要求を行うクエリです。



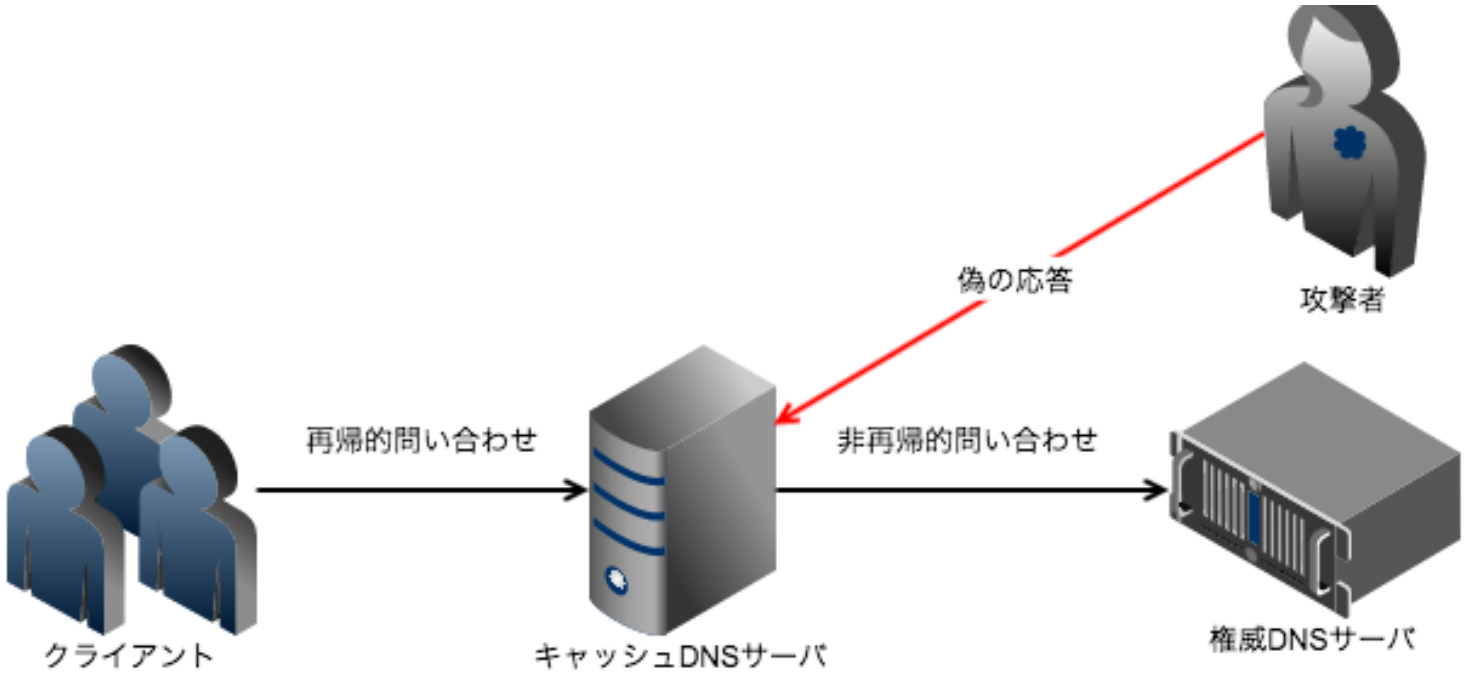
問い合わせの種類について

- 非再帰的問い合わせはキャッシュDNSサーバが権威DNSサーバに対して、名前解決要求を行うクエリです
- キャッシュDNSサーバは再帰的問い合わせをトリガーに非再帰的問い合わせを行います。



問い合わせの種類について

- 毒入れは、非再帰的問い合わせの偽の応答をキャッシュDNSサーバに本物の応答と認識させ、キャッシュさせることで行います。



クエリーの複雑性について

- DNSのクエリは通常UDP Protocolでやり取りを行います。（TCPも使います）
- サーバに問い合わせを行い場合は

  dst port 53です。
- ソースポートは特にきまってません。

  udp ポートは16bit

クエリーの複雑性について

- また、DNSクエリーには16 bit TXID フィールドがあり、これによってDNSクエリーの識別を行います。
- ソースのUDPポートも16bit分の複雑性があります。
- DNSクエリーを識別するには、ポートとTXIDで最大で32bit分の複雑性あります。

権威とRanking Data

- 管理しているゾーンをもつ権威DNSサーバは、そのゾーンの権威を持っています。
- 権威が有る応答の場合はAA bitがたちます

```
; <<>> DiG 9.10.0 <<>> +norec +noedns +nodnssec @d.dns.jp jp. SOA
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43179
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 7, ADDITIONAL: 0

;; QUESTION SECTION:
;jp.    IN     SOA

;; ANSWER SECTION:
jp.     86400       IN     SOA  z.dns.jp. root.dns.jp. 1415088006 3600 900 1814400 900
```

権威とRanking Data

- 委任情報に対して権威はありません。

```
; <<>> DiG 9.10.0 <<>> +norec +noedns +nodnssec @d.dns.jp iij.ad.jp
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34458
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;iij.ad.jp.                     IN      A

;; AUTHORITY SECTION:
iij.ad.jp.              86400   IN      NS      dns1.iij.ad.jp.
iij.ad.jp.              86400   IN      NS      dns0.iij.ad.jp.
```

権威とRanking Data

- そのゾーンの権威DNSサーバのホスト情報は、そのゾーン自体が権威を持ってます。

```
; <<>> DiG 9.10.0 <<>> +norec +noedns +nodnssec @dns0.iij.ad.jp iij.ad.jp ns
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13747
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 4

;; QUESTION SECTION:
;iij.ad.jp.                     IN      NS

;; ANSWER SECTION:
iij.ad.jp.              86400   IN      NS      dns0.iij.ad.jp.
iij.ad.jp.              86400   IN      NS      dns1.iij.ad.jp.
```

権威とRanking Data

- クエリーの応答は、３種類の種類があります。
- ANSWER SECTION
  - 問い合わせに対しての回答。
- AUTHORITY SECTION
  - 委任情報の応答や、不在情報の応答
- ADDITIONAL SECTION
  - 付加情報

権威とRanking Data

- 非再帰問い合わせを行った結果、キャッシュをどう利用するかは、Ranking Dataで決まっています。

権威とRanking Data

Ranking Data（一部抜粋）

	ANSWER	AUTHORITY	ADDITIONAL
権威の有る応答(auth)	1 (authanswer)	2 (authauthority)	4 (additional)
権威の無い応答	3 (answer)	4 glue	4 additional,glue

- 委任元のNS、glueは一番優先度が低いです。
- 攻撃者は権威の有る応答のAUTHORITYを使用して攻撃を行うことが多いです。
- なお、同じRanking Dataを持つ応答を受け取った場合の動作は、定義されていません。

本章まとめ

- DNSには完全性の保証が無い。
- 再帰的問い合わせをトリガーにして非再帰的問い合わせが発生する。
- DNSクエリーの複雑性
  - UDP Port 16bit + DNS TXID Filed 16bit = 32 bit
- 委任にはNSレコードとglueレコードを使用する。
- 委任には権威が無い。
- キャッシュで使われる優先順位
  - authanswer > authauthority >> additional,glue

主なポイズニング手法

主なポイズニング手法

- カミンスキー型攻撃
- 第一フラグメント便乗攻撃
- 経路ハイジャックによる攻撃

それぞれの
- 攻撃手法
- 成功確率
- シュミレーション

カミンスキー型攻撃

- 一番メジャーな攻撃手法

攻撃手法(カミンスキー型)

- 攻撃者は毒入れを行いたいドメイン名のゾーンを持った権威DNSサーバを構築します。

攻撃者

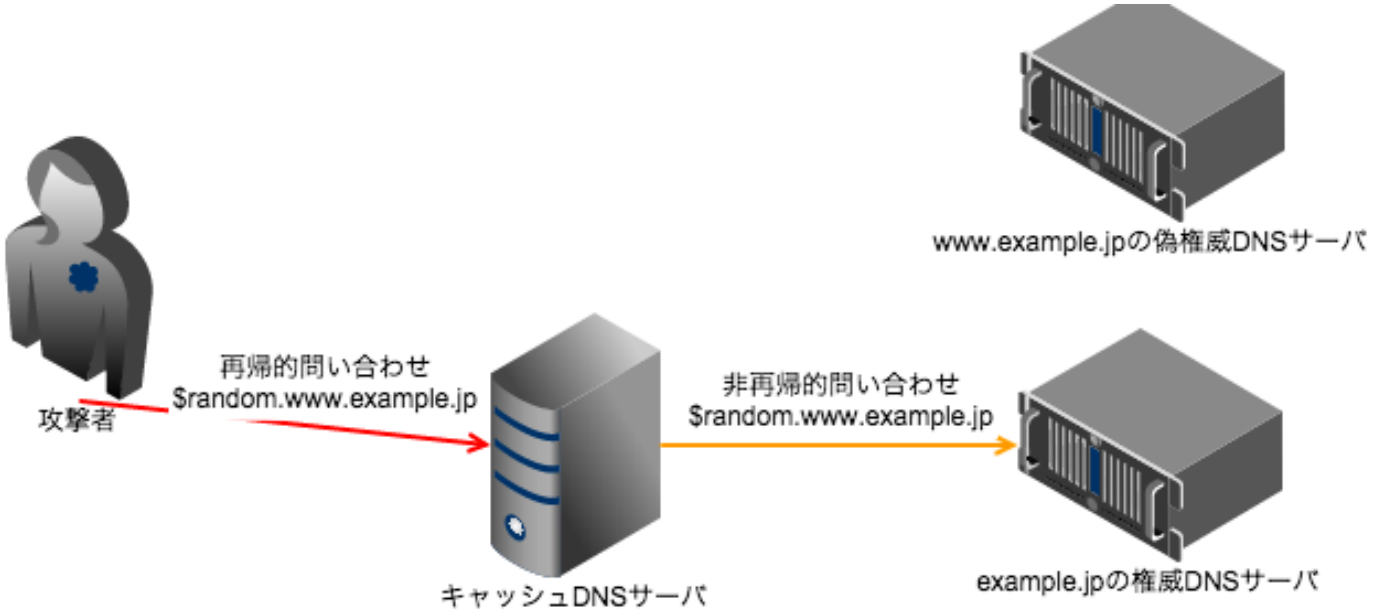
キャッシュDNSサーバ

www.example.jpの偽権威DNSサーバ

example.jpの権威DNSサーバ

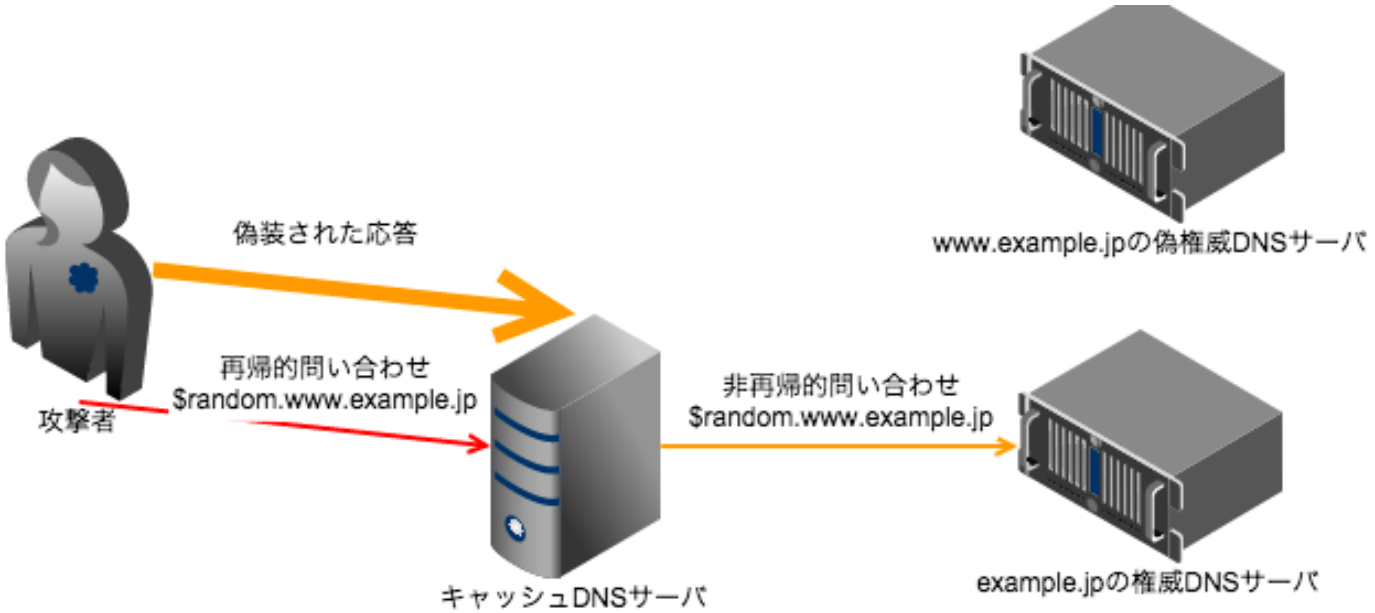
攻撃手法（カミンスキー型）

- 攻撃者がキャッシュDNSサーバに対して、ターゲットドメイン名のランダムサブドメインの再帰的問い合わせを行います。
- (Point)ランダムサブドメインの為、キャッシュに無いため毎回キャッシュDNSサーバは非再帰問い合わせを行います。



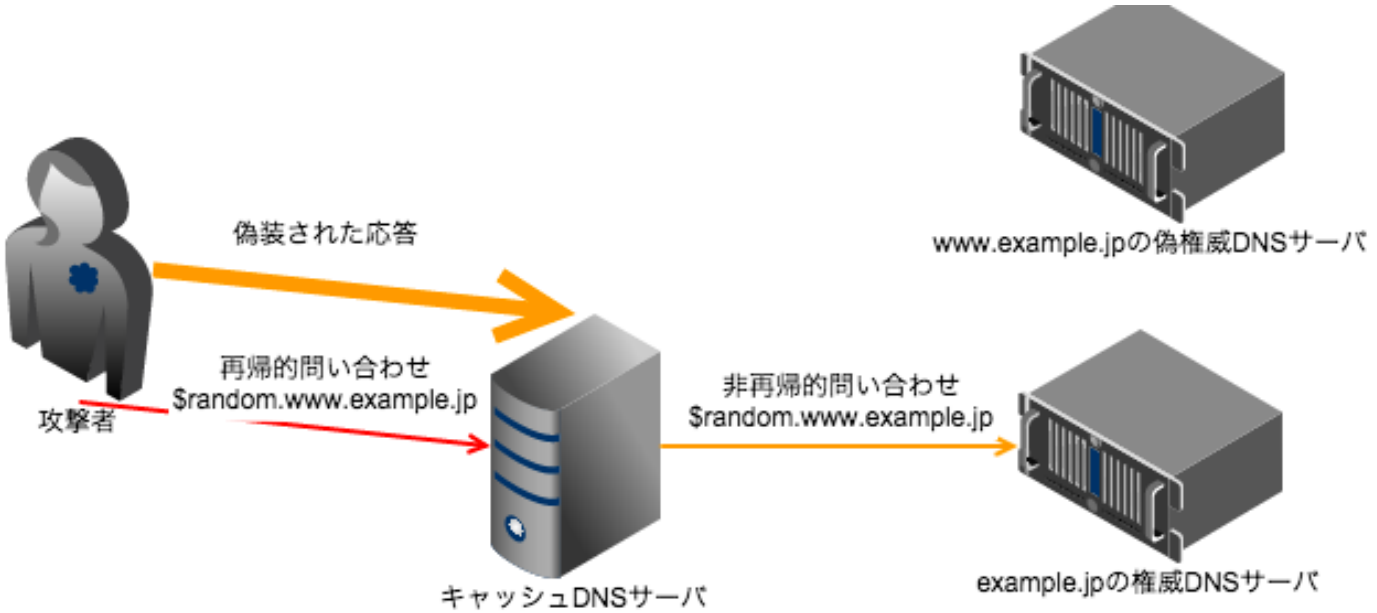
攻撃手法（カミンスキー型）

- 攻撃者が、権威DNSサーバの非再起問い合わせへの応答を偽装したパケットをキャッシュDNSサーバに送ります。



攻撃手法（カミンスキー型）

- 権威DNSサーバの応答より先に、攻撃者のパケットが着信し、UDPポート番号とDNSのTXIDが一致した場合、権威DNSサーバからの応答と見なして、内容をキャッシュします。



攻撃手法（カミンスキー型）

- このとき、authorityのNSレコードを毒入れする事によって、毒入れターゲットのドメイン名をゾーン分割してしまい、偽の権威DNSサーバに誘導します。

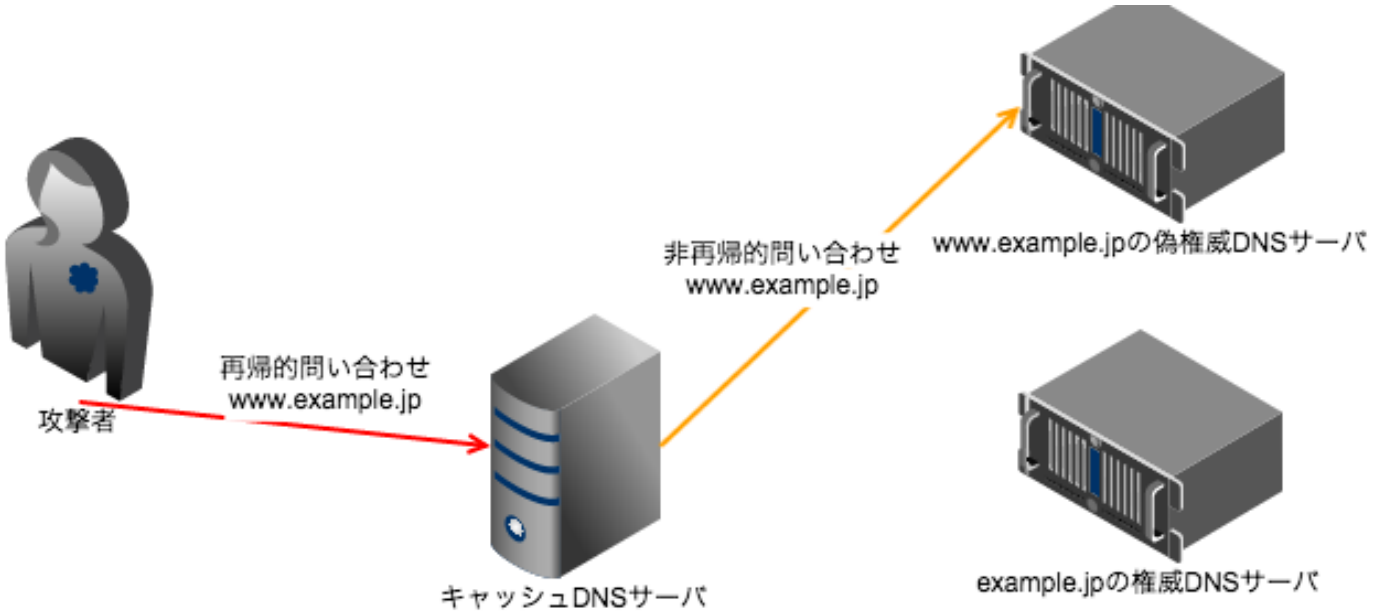
```
;; QUESTION SECTION:
;hogehoge.www.example.jp.              IN     A

;; AUTHORITY SECTION:
www.example.jp.              86400     IN     NS    nom.example.com.
```

攻撃手法(カミンスキー型)

- ターゲットのレコードのキャッシュが消えるのを待ちます。

攻撃手法（カミンスキー型）

- キャッシュDNSサーバに対してターゲット名を問い合わせると、偽の権威DNSサーバから毒入れされたレコードを受け取ります。



攻撃の成功確率（カミンスキー型）

1回の攻撃の成功する確率

$$p_s = \frac{MINUTE \cdot PPS}{ANSIP \cdot CNSIP \cdot TXID \cdot PORT}$$

Ps: 一回の成功確率
MINUTE: 攻撃時間(s)
PPS: 秒間の攻撃パケット数
ANSIP : 権威DNSサーバのIP
CNSIP: キャッシュDNSサーバの非再帰問い合わせに使用するIP数
TXID: DNSヘッダーのTXIDフィールド
Port : 非再帰的問い合わせに使用するUDP Portの数

攻撃シュミレーション（カミンスキー型）

シナリオ（ポート固定）

仮定１）
権威DNSサーバのIP数はほとんどのサーバが2つしかNSを書いてないので、
２個とします。

仮定2)
ポートランダマイズされていないキャッシュサーバなのでUDPPortは1とします。

シナリオ２（ポートランダム）

仮定１）
権威DNSサーバのIP数はほとんどのサーバが2つしかNSを書いてないので、
２個とします。

仮定2)
ポートランダマイズされているキャッシュサーバなのでUDPPortは64000とします。

攻撃成功の時間とPPSの関連

時間内に攻撃成功に必要なPPS

シナリオ	1h	12h	1d	1w	30day	180day	1y
固定	36	3	1				
ランダム	233万	19万	9.7万	1.3万	3236	539	265

時間内に攻撃成功に必要なbps（1pps=512byteで計算）

シナリオ	1h	12h	1d	1w	30day	180day	1y
固定	145K	12k	6k				
ランダム	8G	758M	379M	54M	12M	2M	1M

脆弱な個所（カミンスキー型）

- DNSクエリーを特定する複雑性が、TXID分の16bitしかない。
  - 16bitだと、１時間36pps程度の攻撃で毒入れ可能
  - この程度のppsだと気付く事ができない。
- 不正なデータか検証できない。

第一フラグメント便乗攻撃

- 昨年度話題になった、IPフラグメントを悪用した攻撃手法。
  - IPv6のフラグメント処理でフラグメントIDが推測可能な実装のIPスタックがあるのが原因
  - 要するにセキュリティバグ。
- カミンスキー型に比べて、条件が合えば攻撃の成功確率が高いのが特徴。

攻撃手法（第一フラグメント便乗攻撃）

- 攻撃者は毒入れを行いたいドメイン名のゾーンを持った権威DNSサーバを構築します。

攻撃者

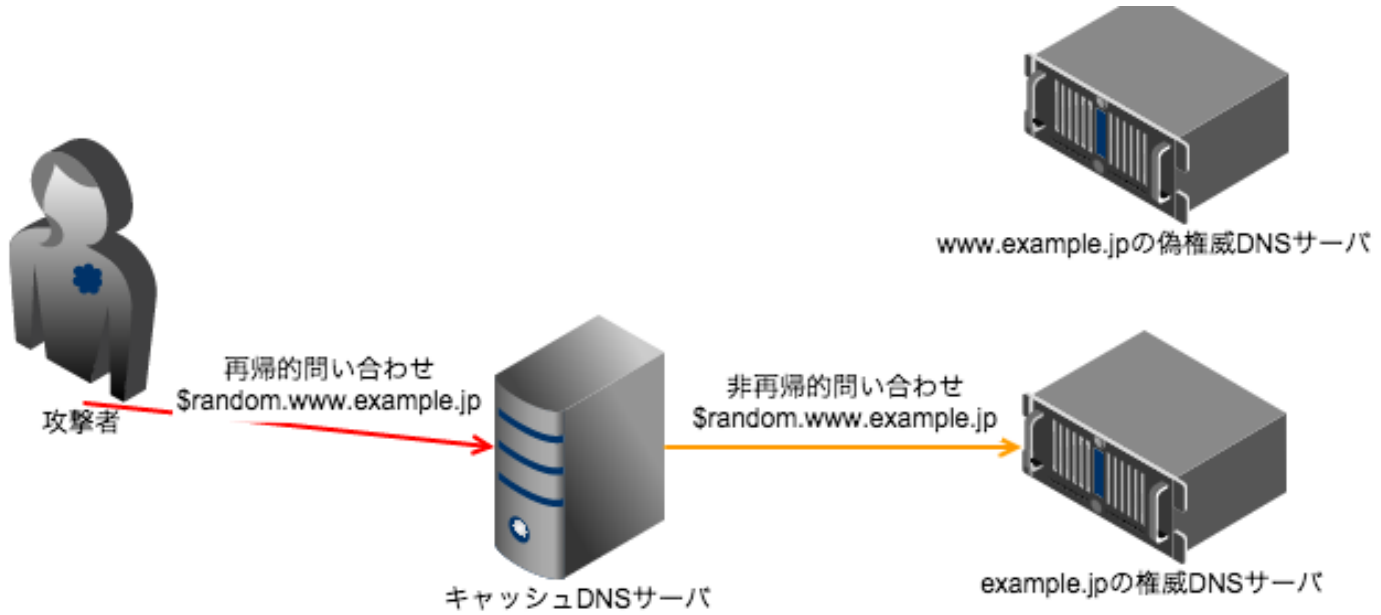
キャッシュDNSサーバ

www.example.jpの偽権威DNSサーバ

example.jpの権威DNSサーバ

攻撃手法（第一フラグメント便乗攻撃）

- 攻撃者がキャッシュDNSサーバに対して、ターゲットドメイン名のランダムサブドメインの再帰的問い合わせを行います。
- (Point)ランダムサブドメインの為、キャッシュに無いため毎回キャッシュDNSサーバは非再帰問い合わせを行います。

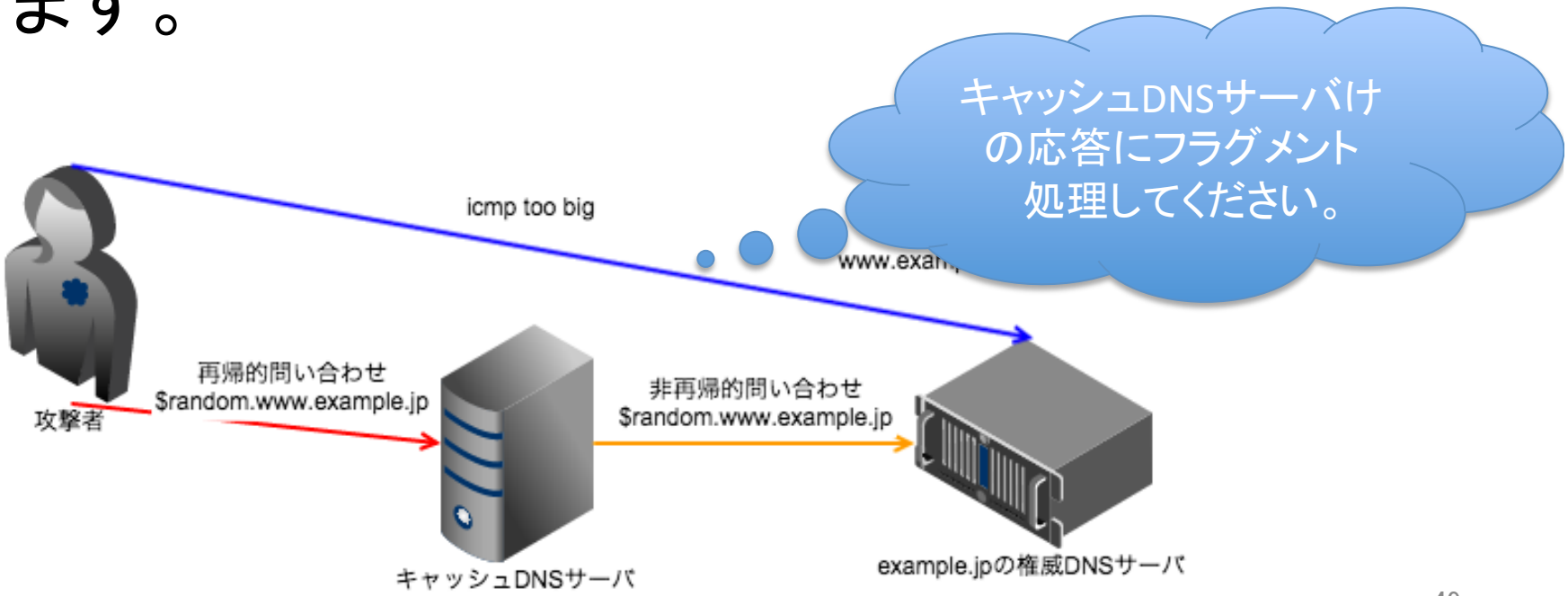


攻撃手法（第一フラグメント便乗攻撃）

- ここまではカミンスキー型と同じ攻撃手法

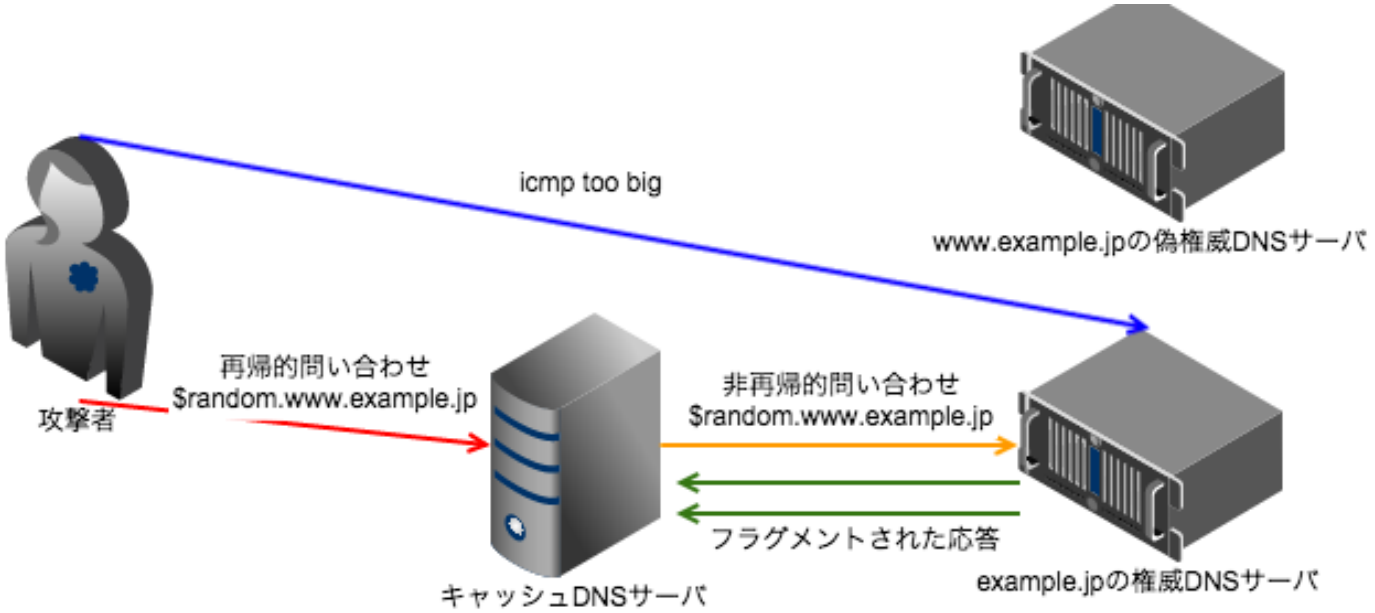
攻撃手法（第一フラグメント便乗攻撃）

- 攻撃者が、権威サーバにキャッシュDNSサーバのIPを使って、icmp too bigを送りつけ、非再起問い合わせへの応答をフラグメントさせます。



攻撃手法（第一フラグメント便乗攻撃）

- 攻撃者が、権威サーバにキャッシュDNSサーバのIPを使って、icmp too bigを送りつけ、非再起問い合わせへの応答をフラグメントさせます。



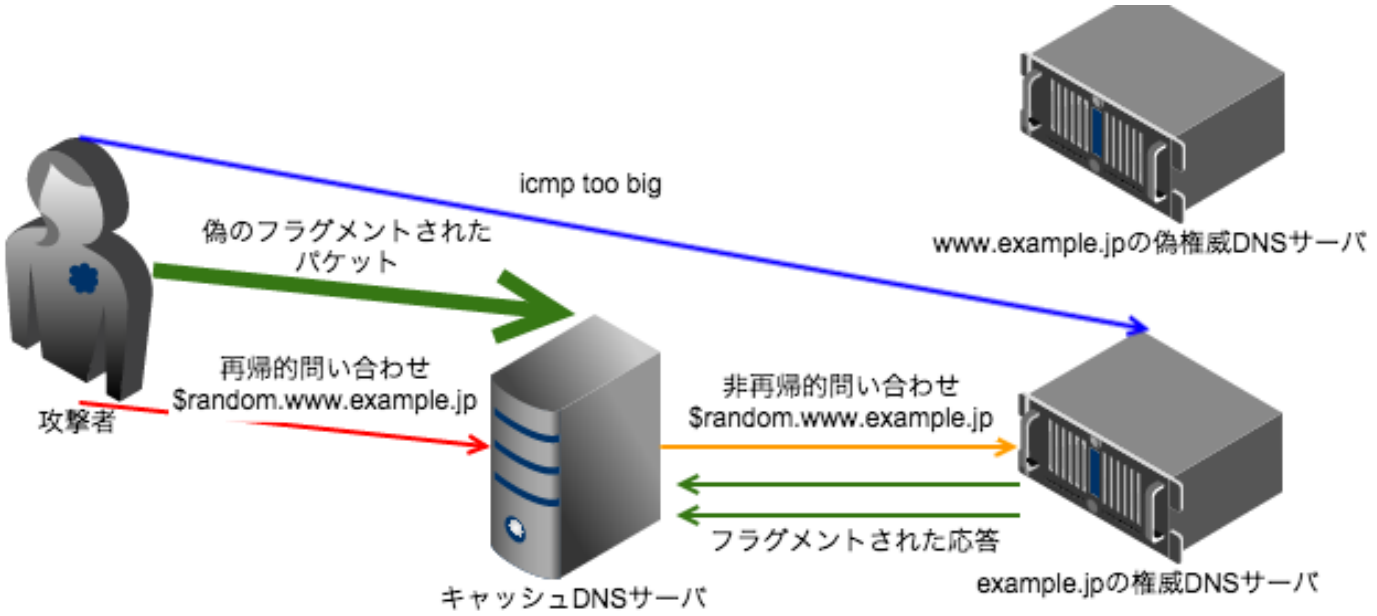
攻撃手法（第一フラグメント便乗攻撃）

- フラグメントすると、先頭のパケット（第一フラグメント）には
  - UDP情報
  - DNS情報のHeader部分
  - が入ると想定できます。

- DNSパケットを認識する為の要素は全部先頭パケットにあります。

攻撃手法（第一フラグメント便乗攻撃）

- 攻撃者がフラグメントで分割された第2フラグメントのパケットをキャッシュDNSサーバに送り攻撃します。



攻撃手法（第一フラグメント便乗攻撃）

- 一方で、フラグメントしたパケットを元に戻す為には、フラグメントIDが一致する必要があります。

- このフラグメントIDが推測可能な場合、攻撃が成立します。

攻撃手法（第一フラグメント便乗攻撃）

- 実際には、それに加えUDPチェックサム(16bit)を合わせる必要があります。

- しかし、応答の形式は有る程度推測できchecksum自体は16bitしか無いため、総当たりで生成することが可能です。

攻撃手法（第一フラグメント便乗攻撃）

- フラグメントIDが一致してしまい、UDPチェックサムも検証できると、キャッシュDNSサーバ側で着信してしまいます。

攻撃手法（第一フラグメント便乗攻撃）

- うまい具合にauthorityのNSレコード偽のフラグメントパケットに組み込み、毒入れする事によって、毒入れターゲットのドメイン名をゾーン分割してしまい、偽の権威DNSサーバに誘導します。

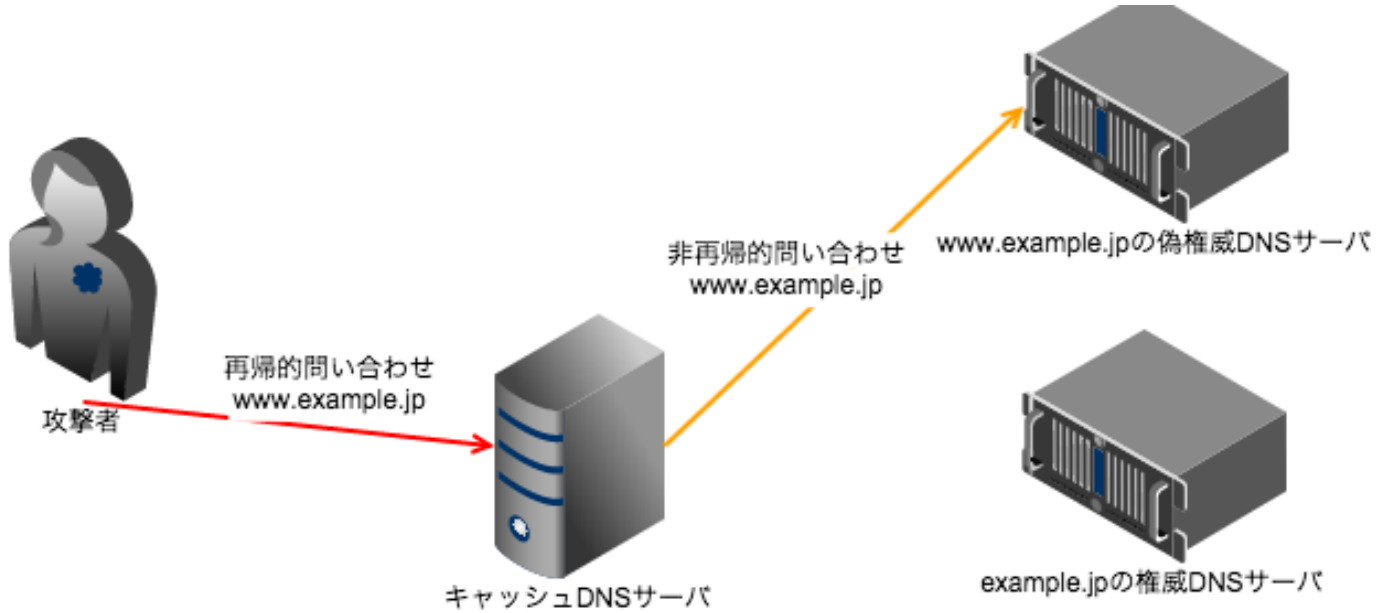
```
;; QUESTION SECTION:
;hogehoge.www.example.jp.              IN     A

;; AUTHORITY SECTION:
www.example.jp.              86400     IN     NS    nom.example.com.
```

攻撃手法(第一フラグメント便乗攻撃)

- ターゲットのレコードのキャッシュが消えるのを待ちます。

攻撃手法（第一フラグメント便乗攻撃）

- キャッシュDNSサーバに対してターゲット名を問い合わせると、偽の権威DNSサーバから毒入れしたいレコードを受け取ります。



攻撃の成功確率（第一フラグメント便乗攻撃）

1回の攻撃の成功する確率

$$P_s = \frac{MINUTE \cdot PPS}{ANSIP \cdot CNSIP \cdot IPID \cdot USUM}$$

Ps:	成功確率
MINUTE:	攻撃可能な時間(s)
PPS:	秒間の攻撃パケット数
ANSIP :	権威DNSサーバのIP
CNSIP :	権威DNSサーバのIP
IPID:	IPのIDフィールド(IPv4:16bit IPv6:32bit)
USUM:	UDP CHECKSUM(16bit)

攻撃のシナリオ

- シナリオ１
  - プロトコルはIPv4
  - 権威DNSサーバの数は2
- シナリオ２
  - プロトコルはIPv6
  - 権威DNSサーバの数は2
- シナリオ３
  - プロトコルはIPv6
  - 権威DNSサーバの数は2
  - フラグメントのIDを推測可能の場合

攻撃成功の時間とPPSの関連

時間内に攻撃成功に必要なPPS

シナリオ	1h	12h	1d	1w	30day	180day	1y
IPv4	2M	194K	97K	13K	3K	552	272
IPv6	145G	12G	6G	887M	207M	34M	17M
IPv6 脆弱性有	36	3	1				

時間内に攻撃成功に必要なbps（1pps=512byteで計算）

シナリオ	1h	12h	1d	1w	30day	180day	1y
IPv4	9G	776M	388M	55M	12M	2M	1M
IPv6	582T	48T	24T	3T	828G	138G	68G
IPv6 脆弱性有	144K	12K	4K				

DNSプロトコルの脆弱性

- 内容の検証ができない。
  - 特に委任情報

経路ハイジャックによる攻撃

- DNSというよりは、BGPとか動的ルーティング側の問題のような気もします。
- HTTPなど、内容の検証ができないプロトコルも同様の脆弱性を抱えています。
  - HTTPS使えってことです。
- DNS側から見ると、正常な応答と何ら変わらないので、DNS側では全く気づけません。

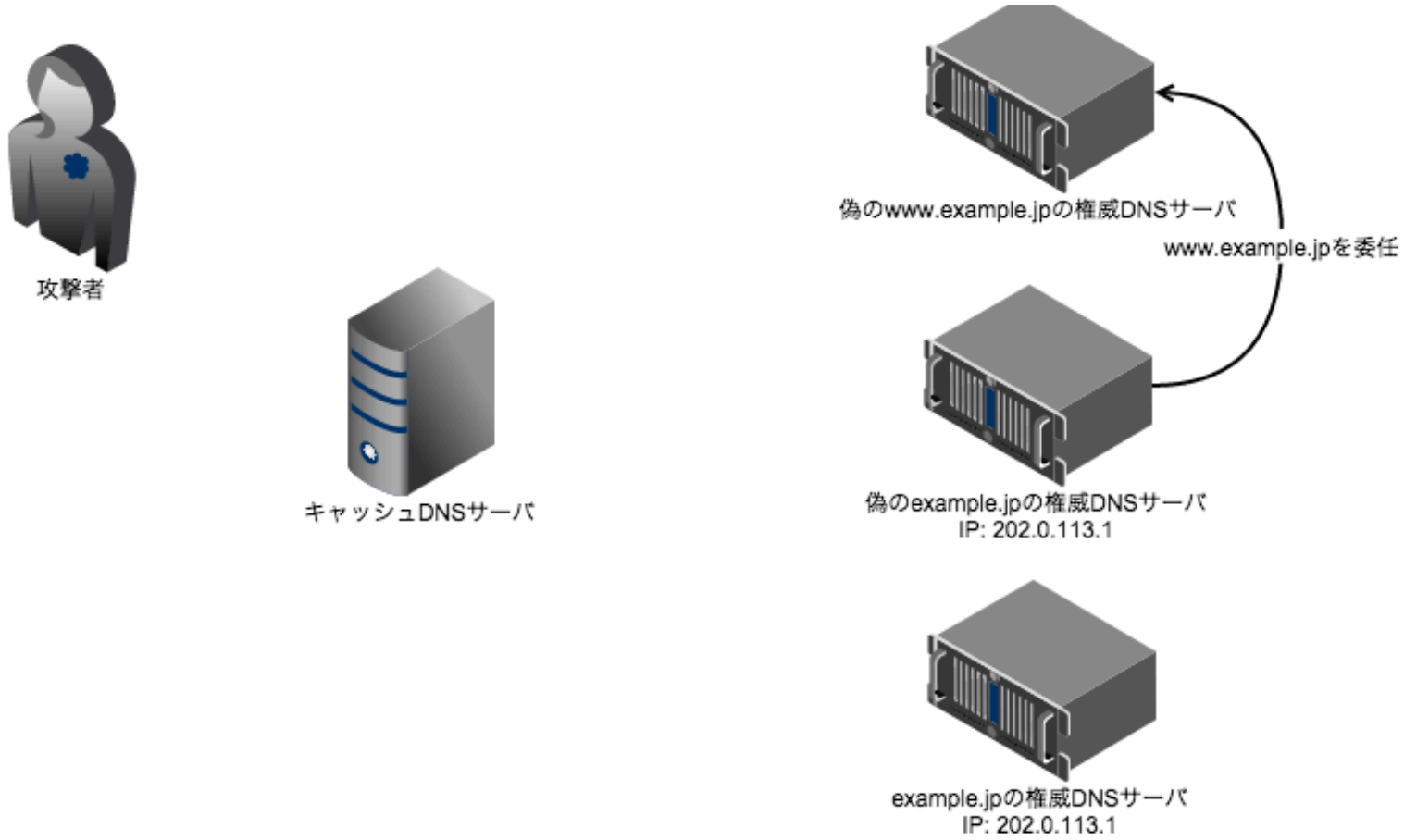
攻撃手法（経路ハイジャックによる攻撃）

- 攻撃者は、ターゲットドメイン名のゾーンを持つ、偽の権威DNSサーバを立てておきます。

攻撃手法（経路ハイジャックによる攻撃）

- 偽の権威DNSサーバへの誘導用に、正規にターゲットドメイン名を管理しているしているゾーンを、別のサーバで立ち上げ、正規の権威DNSサーバと同じIPをつけます。
- この時、ターゲットドメイン名をゾーン分割し、偽の権威DNSサーバに委任します。

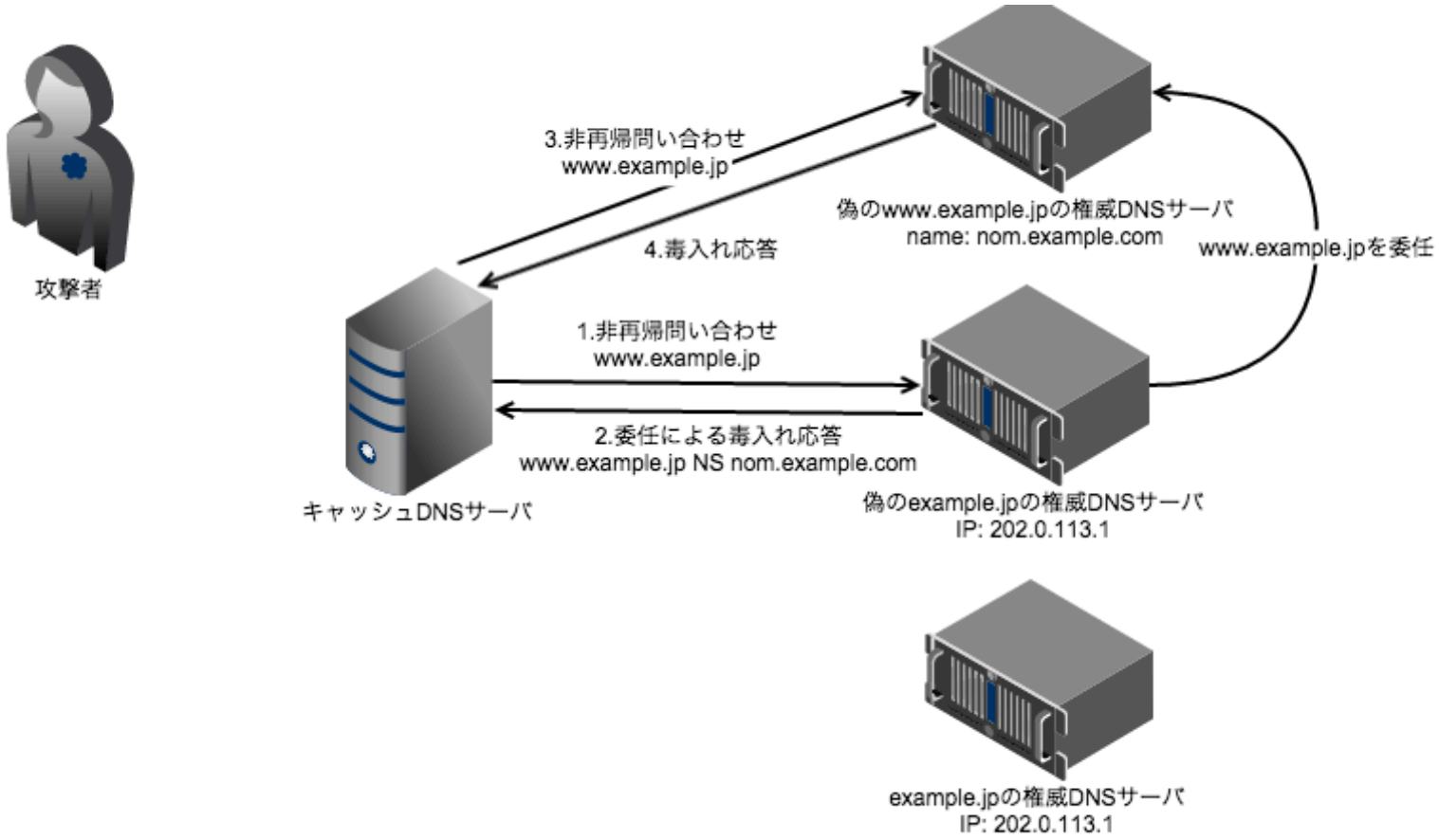
攻撃手法（経路ハイジャックによる攻撃）



攻撃手法（経路ハイジャックによる攻撃）

- 経路ハイジャックを行い、誘導用の権威DNSサーバに非再帰問い合わせが来るようにします。
- この間、経路ハイジャックされたASからの問い合わせを受けると、偽の権威DNSサーバへの委任情報がキャッシュされます。

攻撃手法（経路ハイジャックによる攻撃）



攻撃手法（経路ハイジャックによる攻撃）

- 経路ハイジャックが止まっても、偽の権威DNSサーバの委任情報(NSレコード)のTTLが切れるまで、ターゲットドメイン名は偽の権威DNSサーバに誘導されます。

攻撃の成功確率

- 経路ハイジャックの成功確率によります。
  - 経路ハイジャックに成功したASは全て影響を受けます。

どこが脆弱

- 内容の検証ができない。

本章のまとめ

- 内容の検証ができないDNSは、応答を改竄されると検知ができない。
- DNS プロトコルだけで、毒入れを完全に防ぐのは不可能
- 委任情報への毒入れに非常に弱い
- 毒入れとは確率の問題である。

対策

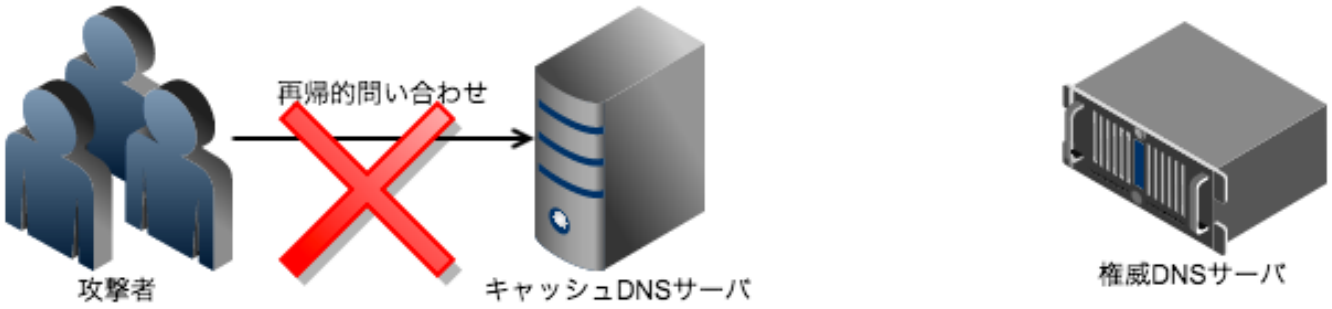
対策

- 攻撃を行いづらくする
- 攻撃の成功確率を下げる
- 被害を局所化する
- 検証できるようにする
- モニタリングする

キャッシュDNS

攻撃を行いづらくする。

- オープンリゾルバを閉じる。
  - 非再帰問い合わせのトリガーになる再帰問い合わせを許可する範囲を制限する。
  - ただし、完全ではないです。
    - 例：網内のbotやDNS Open Proxyの存在



攻撃の成功確率を下げる

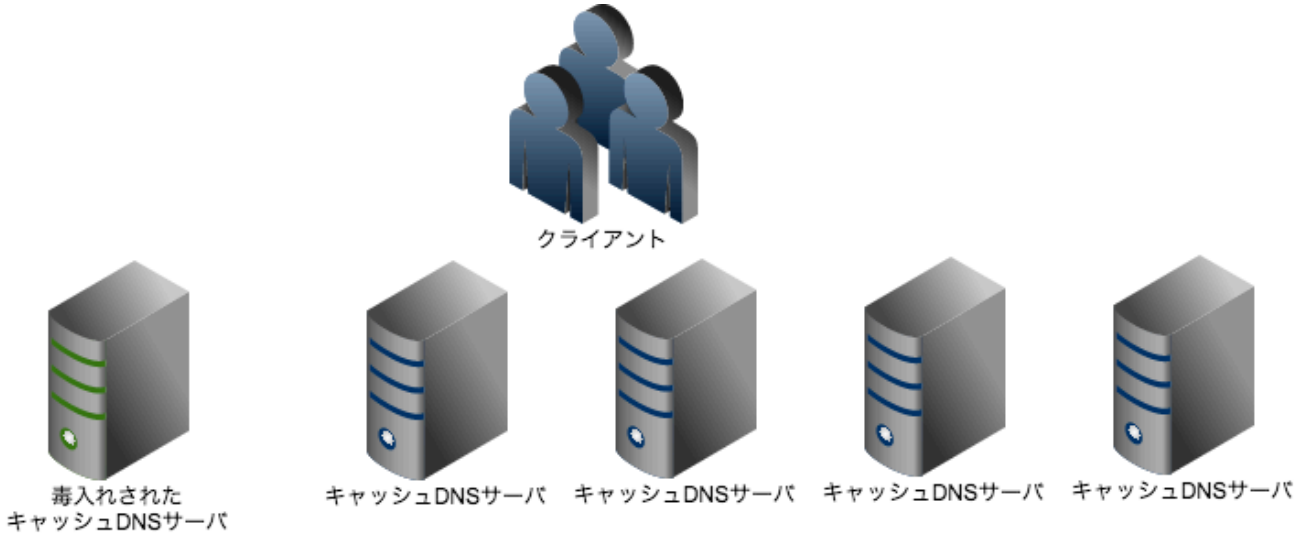
- キャッシュDNSサーバの非再起問い合わせを行う時のポート番号をランダム化する。
  - 何度も出てきてますが、非常に大事です。
  - NAT/NAPTが入ってる場合、ポート変換後のポートがランダムになっていること確認すること。
  - 今使っているDNSの状態を確認したい場合は
    DNS-OARCを使うと便利
    (https://www.dns-oarc.net/oarc/services/porttest)
    dig @server-ip +short porttest.dns-oarc.net txt

攻撃の成功確率を下げる

- キャッシュDNSサーバの非再起問い合わせのソースIPアドレスを複数用意する。
  - 効果として、カミンスキー型の攻撃と、第1フラグメント便乗攻撃の成功確率が、1/IP数に下がる。

被害を局所化する

- LBを用いて複数のサーバで運用する。
- IP Anycastを使い、複数のサイトで運用する。
  - サーバ台数を増やして、毒入れされた場合の影響範囲を小さくする。



検証できるようにする

- DNSSECのValidation機能を有効にする。
  - DNSSECに対応したドメイン名であれば、ほぼ毒入れを防ぐ事が可能。
  - DNSSECに対応していないドメイン名に大しては何ら効果を発揮しない。

モニタリングする

- 対策を施したサーバに対して、キャッシュポイズニングするには大量のパケットが必要
  - 特に、カミンスキー型の攻撃は顕著
  - これを監視することで、攻撃に気づいて、手動でFilterするなどの緊急的な対策が可能となる。
- モニタリングの手段としては、UDPのパケット(pps）が一番有効
- DSCを使って、クエリーのパターンを分析するのも有効な運用法
- これらの値を毎日チェックしておけば、定常状態から外れた場合に気づけます。

権威DNS

攻撃をしずらくする

- IPv6のフラグメントで、フラグメントIDが推測できる実装のOSを使わない。

- 該当するのOSは2012年以前のOSで
  - Solaris10,9,8
    - Solaris10はpatch提供されている。
  - Linux Kernel 3.2より前の実装。
    - 3.2より前でもディストリビューションによってパッチが当てられている。
    - CVE-2011-2699をキーに調べてみればすぐに分かります。

攻撃の成功確率を下げる

- 権威DNSサーバのIP数を増やす。
  - キャッシュDNSがどの権威DNSサーバに問い合わせるか選択しが増えるので、その分攻撃の成功確率が下がる。

被害を局所化する

- 親子同居しない構成にする
  - 親ゾーンが権威を持つ名前に余り問い合わせが来ない場合、親ゾーンのNSがキャッシュされにくい問題が発生するのを防ぐ。

被害を局所化する

- NSの無い名前空間を作らせない設計にする。
  - ゾーンのリーフに「.」を使わない
  - 例
    - example.jp.ゾーンに{www,ftp,mail}.tky.example.jp.のような名前を作る。
    - tky.example.jp部分のNSが無いのでそこに毒入れされる。
    - {www,ftp,mai}がまとめて毒入れできてしまう

検証できるようにする

- 守りたいゾーンをDNSSECに対応させる
  - キャッシュDNSサーバ側で検証してくれれば、ほぼ毒入れを防げるようになる。

対策まとめ

- 対策案は有るのですが、全部やってたらお金がたりません。
- やっておいた方がいいと思う対策を、説明します。

最低限の対策

- キャッシュDNSサーバ
  - キャッシュDNSサーバの非再起問い合わせを行う時のポート番号をランダム化する。
  - オープンリゾルバを閉じる。

- 権威DNSサーバ
  - DNS的に正しい状態になっていること。
    - http://dnscheck.jp/で、検査して問題なければok
    - lameとかキャッシュとの兼用は論外

推奨する対策

- キャッシュDNSサーバ
  - LB構成
  - 監視
  - DNSSEC 検証対応
- 権威DNSサーバ
  - IPv6を使う場合はOSのIPスタック実装に注意
  - 監視
  - DNSSEC対応

推奨する対策の優先順位

キャッシュDNS

対策	優先順位
UDP ポートランダム化	1
オープンリゾルバを閉じる。	2
LB構成	3
監視	4
DNSSEC検証対応	5
Anycast化	6

権威DNS

対策	優先順位
正しい設定	1
IPv6を使用する場合はフラグメントに問題の有る実装を選ばない。	2
監視	3
親子同居	4
NSの無い名前空間を作らせない設計にする。	5
DNSSEC対応	6

まとめ

全体のまとめ

- DNSプロトコルは完全性の保証が無いので、毒入れを完全に防ぐことはできない。
- 対策を施す事により、毒入れのリスクを下げる事は可能。
- 運用をきちんとしていれば、検知も可能になり、リスクはさらに下がる。
- どうしても守りたいドメイン名に対しては、DNSSECを導入する。

付録：用語説明

- 内部名と外部名
  - 自身のゾーン内の名前で有れば、内部名
  - 違うものが外部名
  - 例：
    - www.example.jpはexample.jpの内部名
    - www.example.co.jpはexample.jpの外部名
    - www.example.jpとwww.example.co.jpはjpの内部名

- IP Anycast
  - 同じIPの経路を複数の個所から流すルーティング
  - 経路的に近いところにパケットが吸われる。
  - ROOTや一部のTLDはAnycast化されている。

  - 推奨資料
    - http://www.root-servers.org/
    - Janog34 頑張れIP Anycast

- DSC
  - DNSサーバ上で、パケットキャプチャを行い、見たい情報をグラフ化してくれるオープンソースソフトウェア。
  - 参考資料
    - https://www.dns-oarc.net/oarc/data/dsc

参考資料

- DNS Summer Days2014「wikipedia DNS_spoofingに書かれている攻撃手法の実装と注入にかかる時間の期待値、及び攻撃ツールの最適化について」
- RFC3833「Threat Analysis of the Domain Name System」
- RFC2181「Clarifications to the DNS Specification」