Kaminsky Attackの全て

Kaminsky Attack解説と対策(前編)
DNS DAY ～利用者を守れ～
Internet Week 2008

民田雅人
株式会社日本レジストリサービス

目次

- DNSキャッシュポイズニング(毒入れ)
- 従来型の毒入れ攻撃手法
- Kaminsky型の毒入れ攻撃手法

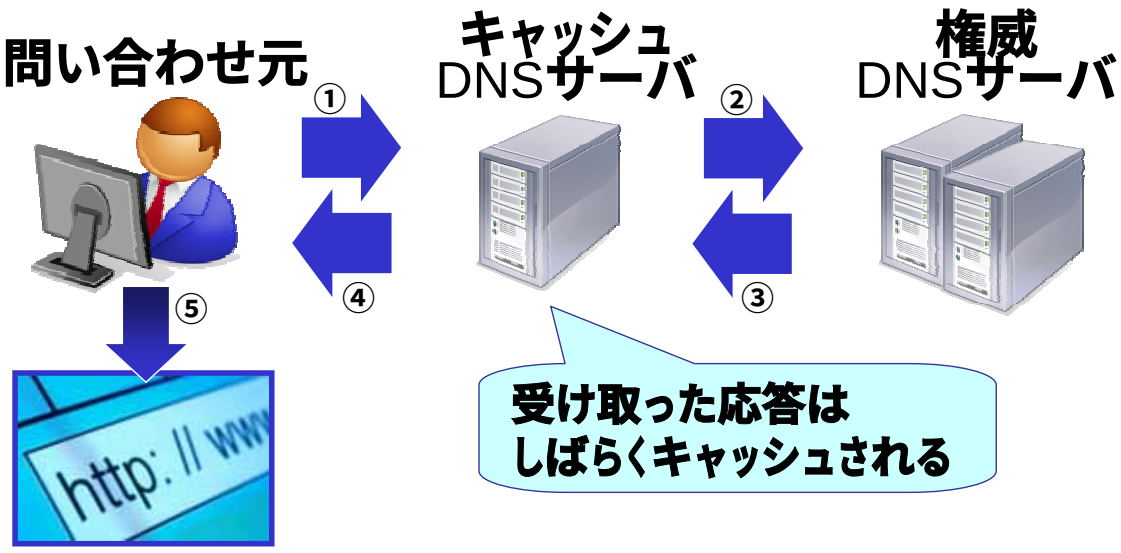
DNSキャッシュポイズニング(毒入れ)

JAPAN REGISTRY SERVICES

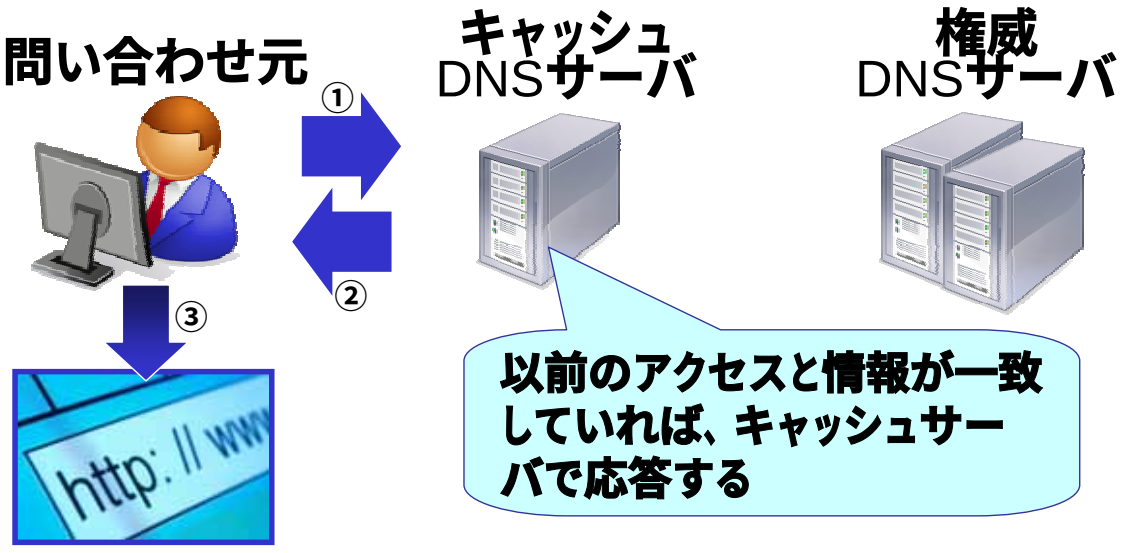
DNSキャッシュポイズニング(毒入れ)

- 予めキャッシュDNSサーバ(以下キャッシュサーバ)に偽の情報を覚えこませ、ユーザが正しいアクセスを行ったつもりでも、偽装サイトへ誘導する手法
- フィッシングやファーミングの為の攻撃手法の一つ

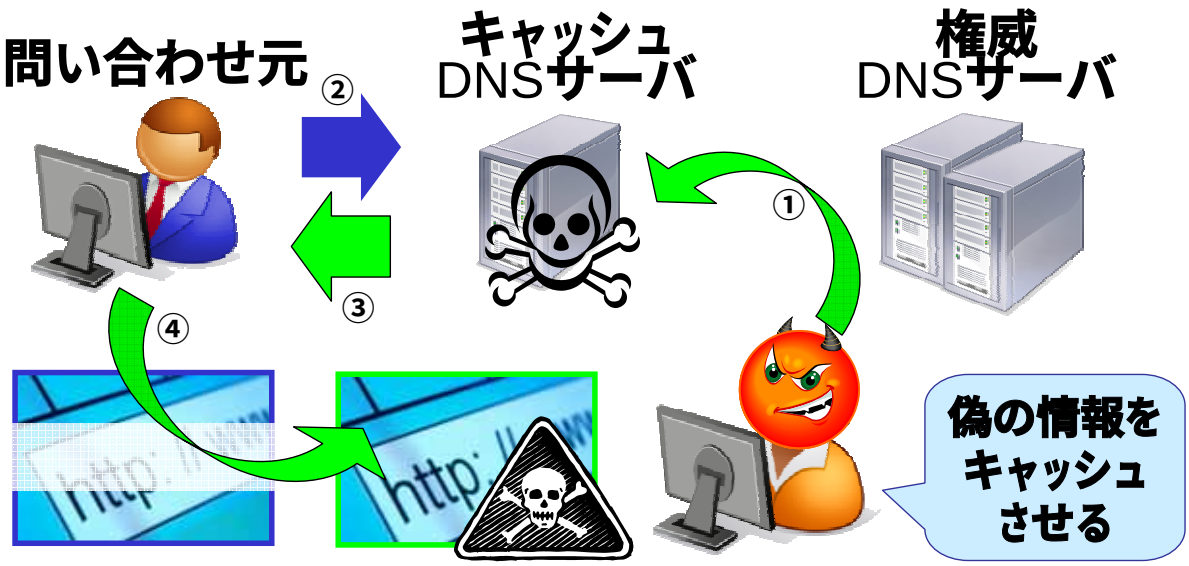
DNSの正常な流れ(1回目のアクセス)



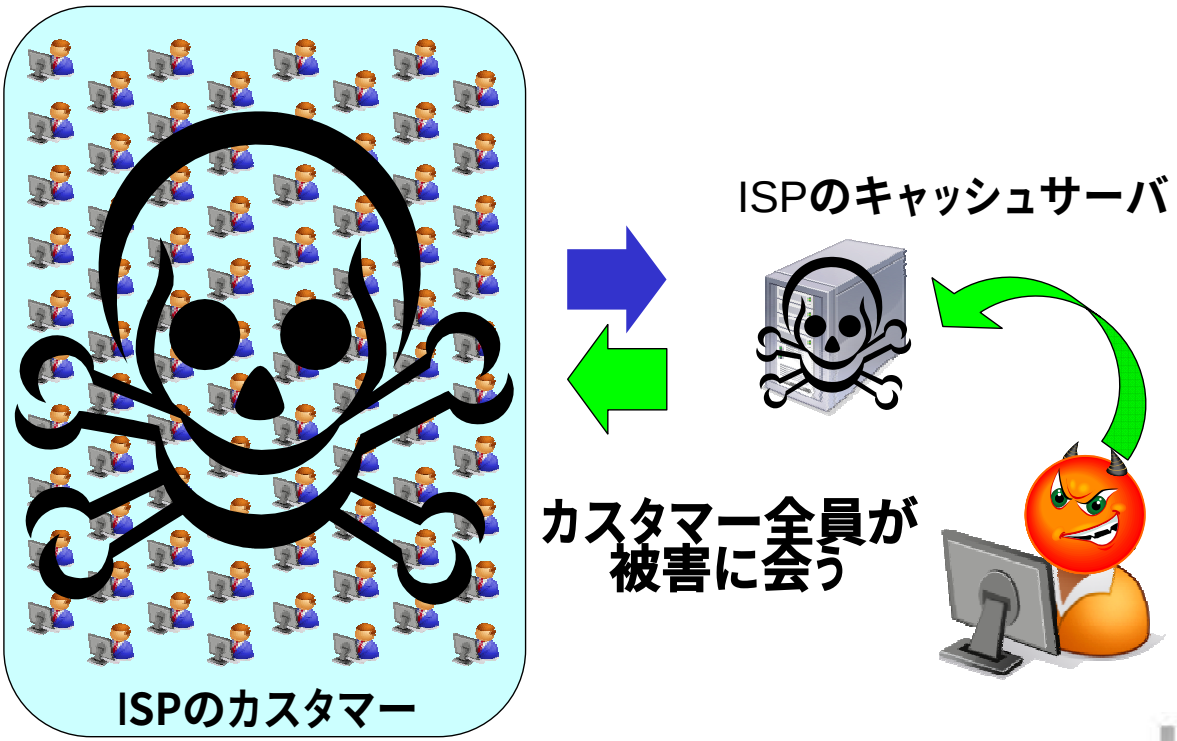
DNSの正常な流れ(2回目以降)



DNSへの毒入れ攻撃



ISPのキャッシュサーバが狙われたら



DNS毒入れ攻撃の特徴

- ユーザが正常なアクセスを行っても、フィッシングサイトに誘導される
  - 攻撃されたことに気づきにくい
- 同じキャッシュサーバのユーザ全員が影響を受ける
  - 大手ISPのキャッシュサーバが攻撃されると被害は甚大
- 攻撃そのものの検出が容易ではない
  - キャッシュへの毒入れは、見た目は通常のDNSパケットであるため、正常な応答と攻撃の区別が簡単ではない

DNSへの毒入れの問題

- 1990年ごろには、DNSへの毒入れの問題が知られていた
  - 当時は設定が正しく行われていないためと考えられていた
  - 攻撃手法として知られるようになったのは1990年代後半
- DNSへの毒入れ攻撃手法の分類
  - Kashpureff型
  - 偽装応答型
  - Kaminsky型

従来型の毒入れ攻撃手法

Kashpureff型による毒入れ

- 攻撃者が管理する権威サーバへ問い合わせさせ、正規の応答パケットに問い合わせ内容と関係ないドメインの情報を附加してキャッシュサーバへ送り込む手法
- 1997年7月の大規模DNS乗っ取り事件
  - http://www.internic.net/ へアクセスすると、http://www.alternic.net/ の内容が表示された
  - AlterNICのEugene Kashpureff氏によるもの

Kashpureff型の攻撃 example.jpでwww.jprs.co.jpの毒入れ

- example.jpゾーンの設定(一般の実装では不可能)

```
@                   IN NS   www.jprs.co.jp.
www.jprs.co.jp.         A   192.0.2.10
www                     A   192.0.2.1
```

- キャッシュサーバがwww.example.jpを検索

```
;; 回答セクション
www.example.jp.         A   192.0.2.1
;; 権威セクション
example.jp.             NS  www.jprs.co.jp.
;; 追加セクション
www.jprs.co.jp.         A   192.0.2.10
```

www.jprs.co.jpの嘘の値をキャッシュする

Kashpureff型攻撃の対策

- キャッシュサーバは、問い合わせたドメインのゾーン外のレコードがあったら、信用せずに捨てる
  - example.jpドメインの応答に、jprs.co.jpの情報があるのはそもそも怪しい

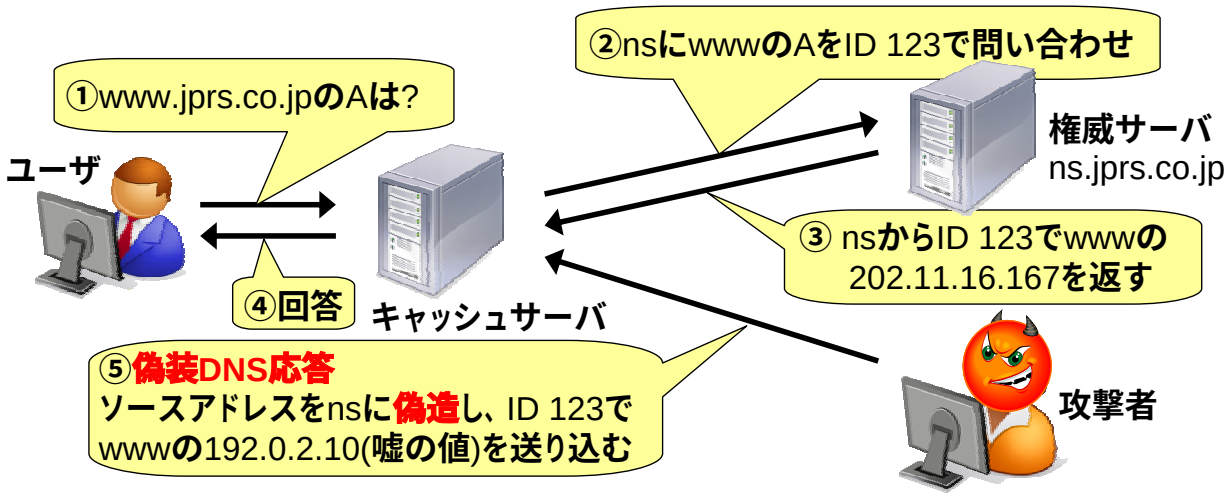
```
;; 回答セクション
www.example.jp.         A   192.0.2.1
;; 権威セクション
example.jp.             NS  www.jprs.co.jp.
;; 追加セクション
www.jprs.co.jp.         A   192.0.2.10 ← 信用してはいけない
```

- BINDの場合4.9.6、8.1.1で対策が行われた
  - 権威サーバ側も対策が行われて設定できなくなった

偽装応答型による毒入れ

- なんらかの手段を使い、本来の応答より先に偽装応答をキャッシュサーバに送り込み、偽情報をキャッシュさせる手法
  - 通常時DNSはUDPで通信するため、偽装応答が容易
- 攻撃手法
  - キャッシュサーバのDNS検索を盗聴し偽装応答を返す
  - キャッシュサーバに問い合わせを送り、IDを変化させた複数の偽装応答を返す(**オープンリゾルバは非常に危険**)
  - TTLの短いレコードを狙って、キャッシュサーバに偽装応答を送り続ける
  - etc...

偽装応答型の攻撃



- ③より先に⑤の偽DNS応答が送り込まれると、キャッシュサーバは嘘情報をキャッシュする
- ④で嘘情報をクライアントに送り、クライアントは偽のサイトへ誘導される

偽装応答型の攻撃が成功する確率

問い合わせと応答のIDが一致すれば攻撃が成功

攻撃1回あたりの成功確率

$$P_S = \frac{R \times W}{N \times Port \times ID}$$

R: 攻撃対象1台あたりに送るパケット量(pps)
W: 攻撃可能な時間(Query⇒AnswerのRTT)
N: 攻撃対象レコードを保持する権威サーバの数
Port: キャッシュサーバのQuery portの数
ID: DNSのID (16bit = 65536)
(*R* 20000pps, *W* 10ms, *N* 2, *Port* 1で 0.00152)

偽装応答型による攻撃の特徴

- 成功確率は決して低いとは言えない
- しかし1度攻撃に失敗すると、キャッシュサーバが正規のレコードをキャッシュするため、連続した攻撃はできない
  - 攻撃に失敗した場合、次の攻撃まで攻撃対象レコードのTTL時間待つ必要がある

Kaminsky型の毒入れ攻撃手法

Kaminsky型の毒入れ攻撃

- 攻撃者がキャッシュサーバに、攻撃対象レコードと同じドメインの**存在しない名前**を検索させ、**追加セクションに攻撃対象レコードを設定**した偽装応答をIDを変化させながら大量に送る(偽装応答型の一種)
- www.example.jpの偽IPアドレスをキャッシュさせる
  - 問い合わせ

```
no0000.example.jp.      A
```

  - 偽装応答

```
;; 回答セクション
no0000.example.jp.      A       192.0.2.1
;; 権威セクション
example.jp.             NS      www.example.jp.
;; 追加セクション
www.example.jp.         A       192.0.2.10
```

Kaminsky型と他の方式の比較

- Kashpureff型との比較
  - 追加セクションを利用する点は同じ
  - Kashpureff型は現在の実装では外部名のため無視されるが、Kaminsky型は内部名となるため、**キャッシュ対象**となる
- 従来の偽装応答型との比較
  - Kaminsky型は**存在しない名前を使用する**ため、攻撃に失敗してもクエリ名を変えることで、TTLに関係なく**連続した攻撃が可能**
    no0000.example.jp, no0001.example.jp, no0002....

Kaminsky型の攻撃はほぼ100%成功する

Kaminsky型攻撃の対策

- 問い合わせポートのランダム化
  - キャッシュサーバの問い合わせポートが固定だったものを、問い合わせ毎にランダムに変化させる
  - 攻撃成功確率を約1/65000に低減できる
- 対処療法ではあるが、実用上問題無い

攻撃1回あたりの成功確率

$$P_S = \frac{R \times W}{N \times 1 \times 65536} \quad \Rightarrow \quad P_S = \frac{R \times W}{N \times 65000 \times 65536}$$

キャッシュ済みのレコードは Kaminsky型の攻撃で上書きできるのか

- 攻撃対象はWEBサーバなどのIPアドレス
  - キャッシュサーバがこのようなレコードをキャッシュしている場合、通常、権威サーバからの正式な回答でレコードを得ている
  - Kaminsky型では、追加セクションにレコードを設定する
- 権威サーバからの正式な回答の方が高ランク
  - RFC2181「5.4.1 Ranking data」より
  - RFCに忠実に実装してあれば、キャッシュデータの上書きは行わない(BIND 9等)
  - RFC通りに実装していない場合、上書きの可能性がある

まとめ

- Kaminsky型の攻撃は、ブルートフォースアタックによる毒入れ攻撃手法
- 未対策かつオープンリゾルバは極めて危険
  - 第三者が自由に攻撃をしかけられる
  - 攻撃対象レコードをキャッシュしていなければ、ブロードバンド回線を使うと10数秒もあれば攻撃が成立
- 毒入れはDNSプロトコルそのものが持つ脆弱性
  - UDPを使う、IDが16bitしかない、etc…
- 完全対処にはプロトコルの拡張等が必要
  - DNSSEC、etc...

Q and A