

DNSセキュリティの現状

～ ルートゾーン キャッシュポイズニングについて ～

<http://www.e-ontap.com/dns/ipsj-tokai2.html>

2014.11.04

情報処理学会東海支部 / 中京大学工学部コロキウム

中京大学工学部 鈴木常彦

今日の話題の経緯

- 2/15 qmail.jp の前野年紀氏のアイデアに基づき、
ゾーンがないサブドメイン名の毒入れに成功し JPRS へ報告
- 2/28 JPゾーン への毒入れに成功し JPRS へ報告
- 3/1 JPRS から JPゾーン への毒入れに別解(親子同居)で成功し、
国内外の関係団体に連絡を取り始めているとの連絡
- 3/16 JPRS が JP下のゾーンがないサブドメイン名に DNSSEC 署名された
TXT レコードを追加 (説明なし)
- 4/1 ルートゾーンへの毒入れ成功し JPRS に報告
- 4/15 JPRS から緊急の注意喚起文書 (問題の解説なし)
- 4/15 解説 「キャッシュポイズニングの開いたパンドラの箱」 を公開

DNSキャッシュポイズニング解説

何が問題なのか？

手法は概ね既知だったが、、、

- 2008年の Blackhat での Kaminsky の説明は誤り
 - Additional Section の毒はそのままでは有効にならない
- 同年の B.Müller の論文が正解
 - "IMPROVED DNS SPOOFING USING NODE RE-DELEGATION", 2008.7.14
 - ノードを偽権威サーバへ誘導する
- 2009年の Blackhat で Kaminsky が再発表
 - Non-existent subdomains can't already be cached, so they're easy to inject.

再委任攻撃の深刻さの指摘 (今回)

キャッシュがない、または入る機会が少ないゾーンが危険

- ゾーンのないサブドメイン名が危険 (co.jpなど)
- 世代間同居のドメイン名が危険
(nic.ukなど / dns.jp は 7/24に解消したが説明なし)
- 子孫の名を NS に用いる親ドメイン名が危険 (nic.uk -> uk)



条件が揃わなくともノード単位で毒は入るが、手間をかけずに広範囲なゾーンを乗っ取ることができる。ルートゾーンまでも。

弱点その1

～ キャッシュが存在しないケース ～

- NS のない (ゾーンが分離されていない) サブドメイン名
 - 例: go.jp, aichi.jp, gouv.fr
 - 上位ドメイン名のゾーンは委任応答 (NS+A) を返さない
(キャッシュは常に存在しない)
 - 存在しない名前には上位ドメイン名のゾーンが否定応答
 - 偽応答で委任を行うと偽権威サーバへ誘導できる

弱点その2

～ キャッシュに入る機会が少ないケース ～

- 世代間同居の子孫側 (人気のないドメイン)
 - 例1: dns.kr (親 kr と子 dns.kr が同居)
 - 子ゾーンの存在しない名前に親ゾーンが否定応答 (するように見えるが実際は子ゾーンが応答)
 - 子ゾーンが引かれる機会が少ない場合、子の応答に含まれる NS+A がキャッシュに入る機会も少ない。
 - 偽応答で委任を行うと偽権威サーバへ誘導できる

```
> dnsq a asfsd.dns.kr a.root-servers.net
```

```
1 asfsd.dns.kr:
```

```
278 bytes, 1+0+6+8 records, response, noerror
```

```
query: 1 asfsd.dns.kr
```

```
authority: kr 172800 NS g.dns.kr
```

```
authority: kr 172800 NS f.dns.kr
```

```
authority: kr 172800 NS e.dns.kr
```

```
authority: kr 172800 NS d.dns.kr
```

```
authority: kr 172800 NS c.dns.kr
```

```
authority: kr 172800 NS b.dns.kr
```

```
> dnsq a asfsd.dns.kr c.dns.kr
```

```
1 asfsd.dns.kr:
```

```
90 bytes, 1+0+1+0 records, response, authoritative, nxdomain
```

```
query: 1 asfsd.dns.kr
```

```
authority: dns.kr 86400 SOA g.dns.kr domain-manager.nic.or.kr 712062200 3600
```

```
900 604800 86400
```

```
> dnsq ns dns.kr c.dns.kr
```

```
2 dns.kr:
```

```
272 bytes, 1+6+0+8 records, response, authoritative, noerror
```

```
query: 2 dns.kr
```

```
answer: dns.kr 86400 NS d.dns.kr
```

```
answer: dns.kr 86400 NS e.dns.kr
```

```
answer: dns.kr 86400 NS g.dns.kr
```

```
answer: dns.kr 86400 NS c.dns.kr
```

```
answer: dns.kr 86400 NS b.dns.kr
```

```
answer: dns.kr 86400 NS f.dns.kr
```


弱点その2

～ キャッシュに入る機会が少ないケース ～

- 世代間同居の子孫側 (子より孫が人気)
 - 例2: `www.foo.bar.internot.jp`
(親 `internot.jp` と子 `bar.internot.jp` が同居)
 - 子ゾーンが引かれる機会が少ない場合、子への委任応答 (NS+A) がキャッシュに入る機会も少ない
 - 孫ゾーンの名前が引かれる場合は、子ではなく孫への委任応答が返る
 - 偽応答で委任を行うと偽権威サーバへ誘導できる

www.foo.bar.internet.jp だけを引く限り、
bar.internet.jp の NS がキャッシュに入ることはない。

```
% dnsq ns bar.internet.jp ns.internet.jp  
answer: bar.internet.jp 3600 NS ns.bar.internet.jp  
additional: ns.bar.internet.jp 3600 A 14.192.44.1
```

```
% dnsq a www.foo.bar.internet.jp ns.internet.jp  
query: 1 www.foo.bar.internet.jp  
authority: foo.bar.internet.jp 3600 NS ns.foo.bar.internet.jp  
additional: ns.foo.bar.internet.jp 3600 A 14.192.44.4
```

```
% dnsq a www.foo.bar.internet.jp ns.foo.bar.internet.jp  
answer: www.foo.bar.internet.jp 7200 A 127.0.0.1  
authority: foo.bar.internet.jp 7200 NS ns.foo.bar.internet.jp  
additional: ns.foo.bar.internet.jp 7200 A 14.192.44.4
```

弱点その3

～ 親の NS が子ゾーンの名 ～

- 子への毒入れで親ゾーンが乗っ取られる
 - dns.kr への毒入れで KR ゾーン
(以前は dns.jp も容易だった)
 - gtld-servers.net への毒入れで NET, COM ゾーン
 - root-servers.net への毒入れでルートゾーン

委任インジェクション

委任インジェクション

- 存在しないサブドメイン名を問い合わせても NS が返らない場合に成立 (親子同居やゾーン非分離)
- 委任元になりすまして、偽委任応答を返し、キャッシュサーバを偽権威サーバへ誘導する
- 下位ノードのキャッシュの存在には影響されない
(`www.example.co.jp` のキャッシュがあっても `co.jp` の委任は可能)

CO.JP ゾーンの乗っ取り

問い合わせ: \$random.co.jp. IN A

jp からの本物の応答: NXDOMAIN

jp からの偽応答:

AA フラグ : 0

Answer Section : なし

Authority Section : co.jp. IN NS ns.poison.nom.

KR ゾーンの乗っ取り

問い合わせ: \$random.dns.kr. IN A

kr からの本物の応答: NXDOMAIN

kr からの偽応答:

AA フラグ : 0

Answer Section : なし

Authority Section : dns.kr. IN NS ns.poison.nom.

(この後、 [b-g].dns.kr に毒を入れると kr が乗っ取れる)

移転インジェクション

RFC2181

- Data from a primary zone file, other than glue data,
- Data from a zone transfer, other than glue,
- The authoritative data included in the answer section of an authoritative reply.
- **Data from the authority section of an authoritative answer,**
- Glue from a primary zone, or glue from a zone transfer,
- Data from the answer section of a non-authoritative answer, and non-authoritative data from the answer section of authoritative answers,
- Additional information from an authoritative answer,
Data from the authority section of a non-authoritative answer,
Additional information from non-authoritative answers.

権威サーバからの NS は 委任元からの NS に優先する

実装によって同ランクのデータを上書きできる

移転インジェクション

- 権威サーバになりすましてネームサーバ情報を上書きし、偽権威サーバへ誘導する
- 権威サーバからのネームサーバ情報が委任元からの情報を上書きする (RFC2181 / 一部無視する実装あり)
- 本物の権威サーバから得たネームサーバ情報も上書きできる (RFC2181は禁止していない / 実装依存)
- 禁止するとネームサーバの移転時の利便性が下がる (NS 移転を真似るのがこの移転インジェクション)

JP ゾーンの乗っ取り

問い合わせ: \$random.jp. IN A

jp からの本物の応答: NXDOMAIN

jp からの偽応答:

AA フラグ : 1

Answer Section : \$random.jp IN A 192.0.2.1

Authority Section : jp. IN NS ns.poison.nom.

実に簡単!

ルートゾーンの乗っ取り

問い合わせ: \$random. IN A

. からの本物の応答: NXDOMAIN

. からの偽応答:

AA フラグ : 1

Answer Section : \$random. IN A 192.0.2.1

Authority Section : . IN NS ns.poison.nom.

ルートゾーンの乗っ取り

問い合わせ: \$random. IN A

. からの本物の応答: NXDOMAIN

. からの偽応答:

AA フラグ : 1

Answer Section : \$random. IN A 192.0.2.1

Authority Section : . IN NS ns.poison.nom.

これは失敗する

priming

- BIND, Unbound は最初の問い合わせ時に . の NS を問い合わせてキャッシュに入れる
- さらに最近の BIND はルートゾーンへ問い合わせが発生するたびに . の NS を問い合わせる (検証用?)
- . の権威ある NS が常にキャッシュにあるため、これを上書きするのは困難

root-servers.net の乗っ取り

- root-servers.net と . は親孫同居
- net の NS がキャッシュになれば委任インジェクションが可能、、、しかし、それは非現実的
- 以下の移転インジェクションが可能

問い合わせ: \$random.root-servers.net. IN A

本物の応答: NXDOMAIN

偽応答:

AA フラグ : 1

Answer Section: \$random.root-servers.net IN A 192.0.2.1

Authority Section: root-servers.net. IN NS ns.poison.nom.

ルートゾーンの乗っ取り

```
; Auth Authority
```

```
. 1111 NS [a-m].root-servers.net.
```

```
; Auth Authority
```

```
root-servers.net. 3333 NS ns.poison.nom. (偽権威)
```

```
root-servers.net. 2222 NS [a-m].root-servers.net.
```

```
; glue
```

```
[a-m].root-servers.net. 2222 A 192.0.2.1
```

この状態で偽権威へ問い合わせた [a-m].root-servers.net の応答は
キャッシュを authanswer で上書きする。ルート乗っ取り完了。

```
; authanswer
```

```
[a-m].root-servers.net. 4444 A 192.0.2.2
```


对策

DNSSEC は対策になるか

- 委譲の Authority Section は検証されない
- 偽権威に誘導され偽 NS キャッシュの TTL が切れるまで DoS 状態
- OPT-OUT運用だとどうなる?
 - RFC5155 (Errataもあり難解)
 - `gouv.fr` は opt-out 運用
 - `www.example.gouv.fr` は守られていないかも
 - `co.jp` も opt-out 運用だが (TXTの役割の意味...)
 - `www.example.co.jp` は守られていた (他の署名済ドメインの存在)
 - `www.example.aichi.jp` は守られていなかったかも
- D.J.Bernstein が 2009年に気づいていた

Breaking DNSSEC

D.J.Bernstein, 2009

Easiest, most powerful attack:

Can ignore signatures.

Suppose an attacker forges a DNS packet from .org,

including exactly the same DNSSEC signatures but changing the NS+A records to point to the attacker's servers.

Breaking DNSSEC

D.J.Bernstein, 2009

Fact: DNSSEC “verification”
won’t notice the change.

The signatures say nothing about the NS
+A records.

The forgery will be accepted.

キャッシュサーバでの対策

- アクセス制限を行う (オープンリゾルバは危険)
- ポートランダムマイゼーション (ポート固定は非常に危険)
- 0x20 などによるエントロピー増加
- 再検査など実装でのアドホックな対応
- 攻撃の検知とキャッシュクリア
- EDNS0 をやめて TCP を使用
(第一フラグメント便乗攻撃対策にも)
- ルートサーバを自前で用意

検知は容易か？

- query を監視？
 - mix7a3pwf9v.jp のような query を監視？
 - example.jp, exbmples.jp, excmple.jp....
みたいな query だったら？
(これでも jp への毒入れ可能です)
- response を監視しましょう

コンテンツサーバでの対策

- NSなしノードの解消
- 親子同居構成の解消
 - 大学等では委任元がセカンダリを引き受けているケースが多い
- NS名の見直し (親が子ゾーンの名を使わない)
- TCPを提供する (もともと要件)
- Lame delegation をなくす
 - 保守でも長期停止は危険

根本対策

- RFC2181 の見直し (可能?)
 - 権威サーバによる NS キャッシュ 書き換えの制限
→ NS 移転に影響
 - 権威より委任元を優先?
 - 議論は DNS の「委任」の設計自体にいきつく
- ZONE Apex(頂点) への Aレコードを禁止
(無理でしょう)
- DNS を捨てる

検証環境

- FreeBSD RELEASE 9.1 -STABLE
 - VIMAGE カーネル (jail vnet)
 - DUMMYNET (遅延用)
 - 仮想ネットワーク構築ライブラリ VITOCCHA (自家製)
 - BIND 9.9.2-P2, Unbound 1.4.20, NSD3 他
- 仮想サーバ群
 - ルートサーバ、TLD サーバ、SLDサーバ (NSD)
 - 偽権威サーバ (NSD)
 - 攻撃サーバ (Metasploit +自作モジュール)
 - キャッシュサーバ (BIND, Unbound)

現状

- 良い状況ではないと思います
 - 公的機関がまだ解説付きの注意喚起をしていない (JPRS, JPCERT/CC, IPAなど半年以上放置)
 - 我々 (前野・鈴木) はなぜ公表したか
 - > JPRS が CO.JP 等に対策し、
説明なしに2008年と同様の注意喚起を行った。
 - > 攻撃者は理由を考える
 - > 攻撃方法を見つけ出す
 - > 防御側がわかっていないのはまずい

現状

- 長らく適切な警告無く放置状態
 - 2008年 Blackhat で Kaminsky が誤った説明 (信じて騒ぎに)
 - 2009年 Blackhat で Kaminsky が正しい説明 (話題にならず) さらに存在しない名前が危ないことを警告
 - 2009年冬に Kaminsky バグがひっそり修正
 - 5年 (Kaminsky 2度目) ~ 6年 (Kaminsky/Mueller) 間、正しい解説がなされず放置
 - 責任ある組織たちは無能だったのか隠蔽していたのか (ハンロンの剃刀)