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

Kazunori Fujiwara, JPRS
fujiwara@jprs.co.jp
2014/6/27

本日の目的と範囲

- 攻撃の原理をわかりやすく伝えることで理解を深め、対策を考えるうえでの参考とする
- 攻撃にかかる時間の期待値を示すことで、正しく運用していれば問題がないことを示す
- RFC 2181のランキングの問題については触れない
  - 今後IETFで動きます
  - フルリゾルバを作りたくなってきた

Copyright © 2014 Japan Registry Services Co., Ltd.

DNS spoofing

- Wikipedia英語版：
  http://en.wikipedia.org/wiki/DNS_spoofing
- 何年も前に書かれていたらしいので引用

> Redirect the target domain's nameserver
>
> The first variant of DNS cache poisoning involves redirecting the nameserver of the attacker’s domain to the nameserver of the target domain, then assigning that nameserver an IP address specified by the attacker.

- 委任応答を偽造する

Copyright © 2014 Japan Registry Services Co., Ltd.

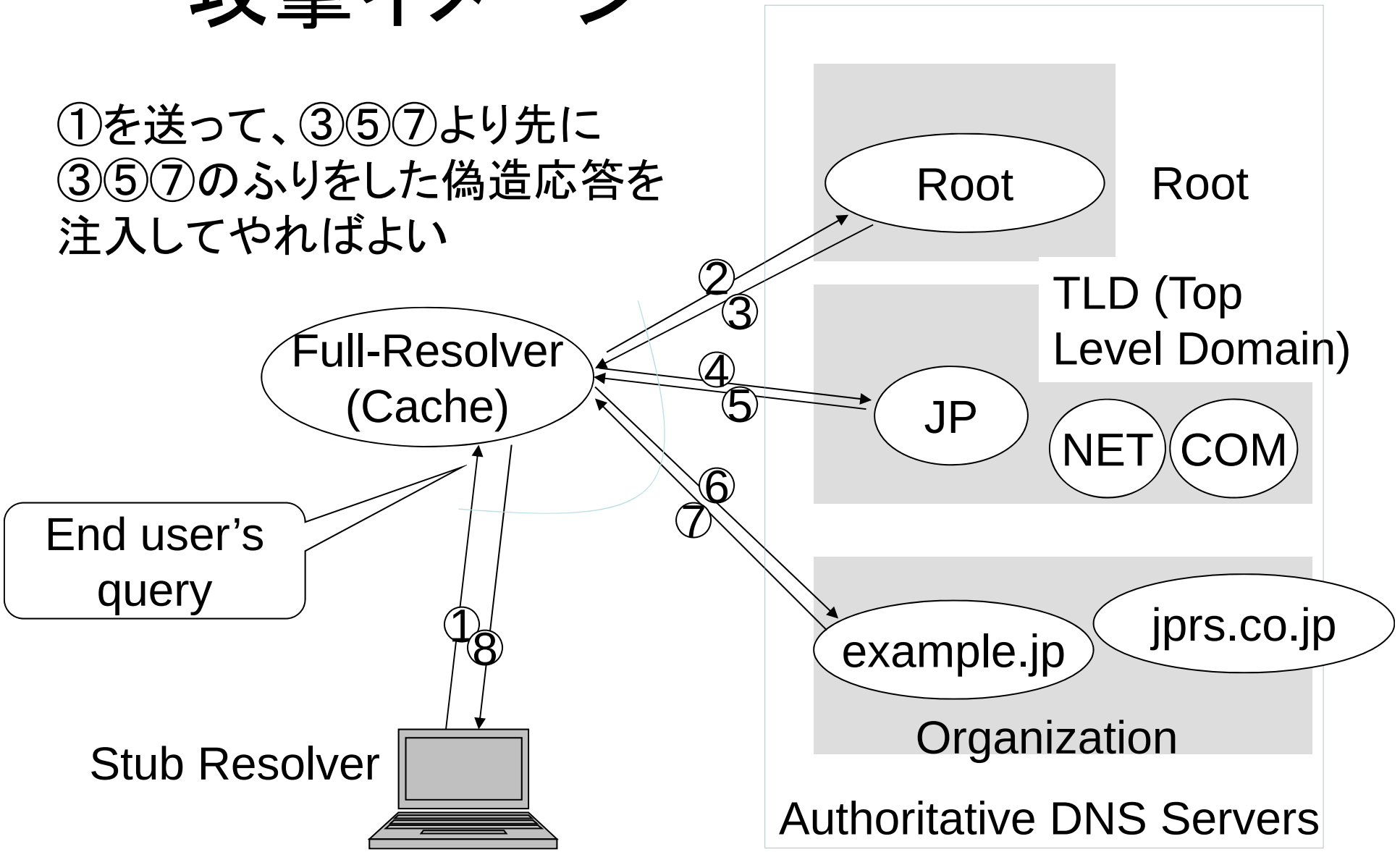
DNS spoofing

- 英語版Wikipediaの該当エントリに「対策はPort RandomizationとDNSSEC」と書かれている
  - 通信路の暗号化については、最近のIETF DNS privacyの議論を先取りしている
- 攻撃手法・対策ともに既知
- RFC 3833の「2.2. ID guessing and query prediction」に分類されるもの
  - IDの推測と問合せの予測
  - RFC 3833：Threat Analysis of the Domain Name System (DNS) (DNSの脅威の分析)

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃イメージ

①を送って、③⑤⑦より先に
③⑤⑦のふりをした偽造応答を
注入してやればよい



Copyright © 2014 Japan Registry Services Co., Ltd.

Kaminsky型攻撃手法

- 攻撃対象の名前の前にランダムなラベルをつけたクエリ名を使用
  - 攻撃対象のサブドメイン
  - 攻撃対象を管理する権威DNSサーバーから返る正当な応答は名前エラー(NXDOMAIN)
  - 正当な応答が先に到達した(攻撃失敗)場合、クエリ名のエラー情報だけがキャッシュされる
    - 攻撃対象の注入には影響がない
  - このため、一度攻撃に失敗してもランダムなラベルを変更することで、連続して攻撃できる

Copyright © 2014 Japan Registry Services Co., Ltd.

Kaminsky型攻撃手法

- ランダムなラベルをつけたクエリ名はキャッシュされていないため、フルリゾルバはクエリを受信するごとに権威サーバにクエリを送る
- これに対する応答を偽造して先に返す
- Kaminsky型攻撃手法では、攻撃者が以下の双方を送ることがポイント
  - ランダムなラベルをつけたクエリ名
  - そのクエリに対応する、権威DNSサーバーからの応答のふりをした偽の応答

Copyright © 2014 Japan Registry Services Co., Ltd.

Kaminsky型の偽装応答攻撃

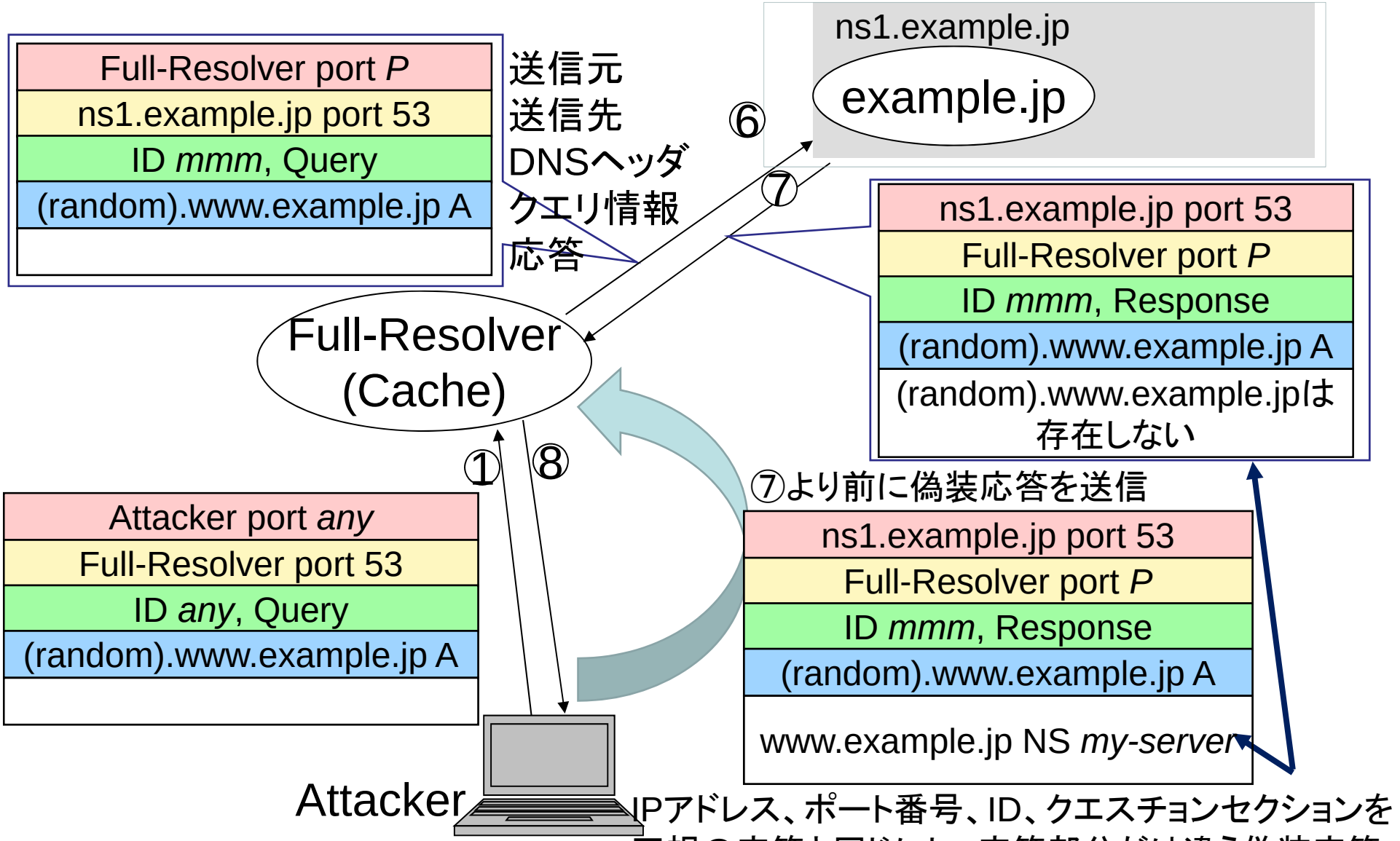
例：www.example.jp というホスト名を注入する場合

1. (random).www.example.jp Aクエリを攻撃対象に送る
   - example.jpを管理する権威DNSサーバは名前エラーを返す
   - 偽造応答の注入が成功しなかった場合、
     - (random).www.example.jpのエラーのみキャッシュされる
     - www.example.jpの情報はキャッシュされない
2. 偽造応答”www.example.jp IN NS *my-server*”をexample.jpのサーバのアドレスから注入する
   - my-serverにwww.example.jp zoneを事前準備しておく

実際の攻撃では、ID(問合せと応答を紐付ける番号)のみを変更した、多数の偽造応答を同時に送る

Copyright © 2014 Japan Registry Services Co., Ltd.

Kaminsky型の偽装応答攻撃



Copyright © 2014 Japan Registry Services Co., Ltd.

対象ドメイン名

- ランダムラベルを前置すると存在しない名前になるドメイン名を「トリガードメイン名」とする
  - (random).www.example.jpの場合、www.example.jpがトリガードメイン名となる
- トリガードメイン名の権威DNSサーバが管理するドメイン名のうち、以下の条件をともに満たすものを注入可能
  1. トリガードメイン名とその祖先のドメイン名
  2. 最も短いゾーン頂点ではないドメイン名

Copyright © 2014 Japan Registry Services Co., Ltd.

対象ドメイン名(ほとんどのホスト名)

例1:トリガードメイン名がwww.example.jpである場合
www.example.jpはexample.jpの権威DNSサーバーが管理しており、この権威DNSサーバーは他のゾーンを一つも管理していないものとする

- www.example.jpを注入できる
- example.jpは「管理するドメイン名のうち最も短いゾーン頂点」であるため注入できない
- jpや“.”は管理していないため注入できない

Copyright © 2014 Japan Registry Services Co., Ltd.

対象ドメイン名(いくつかの中間ドメイン名)

例2:トリガードメイン名がroot-servers.netである場合
(ルートサーバーは“.”とroot-servers.netを管理しており、netの委任情報を持つ)

- root-servers.netを注入できる
- netは「トリガードメイン名の祖先」であるため注入できる
- “.”は「管理するドメイン名のうち最も短いゾーン頂点」であるため注入できない

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃成功確率と期待値の計算

- 作ってみただけでは格調高くないので、モデル化
- モデル化に用いた変数

	変数	単位	数値の範囲	www.google.comでの実績
フルリゾルバのポート数	$Nport$		$1\sim2^{16}$	1
QID数	$Nqid$		2^{16}	65536
権威サーバアドレス数	Nns		$1\sim13$	4
権威サーバへのRTT	$Tauth$	秒	$0.001\sim0.2$	0.039
繰り返し時間	$Tloop$	秒		
偽応答数の送出レート	$Rans$	パケット/秒		

Copyright © 2014 Japan Registry Services Co., Ltd.

一度目の試行で入る確率

- ID, port, アドレスが一致すると確実に入るとする
- 一度の試行($Tauth$時間)で入る確率 P

$$P = \frac{Tauth * Rans}{Nqid * Nport * Nns}$$

- ただし$Tauth$は制御できないので
  $Tloop < Tauth$という条件で$Tloop$を用いる

$$P = \frac{Tloop * Rans}{Nqid * Nport * Nns}$$

Copyright © 2014 Japan Registry Services Co., Ltd.

連続攻撃時に入る確率

- 繰り返せば入るので1になることを示す
- $n-1$回目までに入らなくてn回目に入る確率

$$Pn = (1-P)^{n-1} * P \qquad (1)$$

- n回目までに入る確率QnはPnの和

$$Qn = \sum_{i=1}^{n} (1-P)^{i-1} P \qquad (2)$$

- 変形すると

$$Qn = 1 - (1-P)^{n} \qquad (3)$$

- nを無限大に近づけると

$$Qn \to 1$$

Copyright © 2014 Japan Registry Services Co., Ltd.

連続攻撃時に入る回数の期待値

- $n-1$回目までに入らなくてn回目に入る確率

$$Pn = (1-P)^{n-1} * P \qquad (1)$$

- n回目までに入る期待値Enは回数*確率の和

$$En = \sum_{i=1}^{n} i(1-P)^{i-1}P \qquad (2)$$

- 変形すると

$$En = \frac{1}{P} - \frac{(1+nP)(1-P)^n}{P} \qquad (3)$$

- nを無限大に近づけると

$$En \to \frac{1}{P}$$

成功する場合の期待値なので$0 < P \leq 1$
Enの(3)右辺の第二項は常に正
かつ(2)よりEnは単純増加なので、
Enは$1/P$を上界とする単調増加数列

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃にかかる時間の期待値

$$T = Tloop * E = \frac{Tloop}{P}$$
$$= \frac{Nqid * Nport * Nns}{Rans}$$

- サーバ数4とし、100000ppsで偽装応答を送ると
- Port randomizationしていない場合
  65536*1*4/100000 = 2.62秒
- Port randomizationしていると
  65536*64000*4/100000= 167772.16秒(約2日)
- 2日間も10万ppsの怪しいパケットがくれば気がつくでしょう

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃ツール試作

- Cで500行、標準ライブラリのみ使用
  - selectでタイミングと送受信の管理
  - raw socketで偽造レスポンス送出
- 与えるパラメータ
  - 攻撃対象のフルリゾルバのIPアドレス、ポート番号
  - トリガードメイン名 (連続攻撃向けのドメイン名)
  - 注入したいNS RRのドメイン名、サーバ名
  - トリガードメイン名の権威サーバのIPアドレスリスト
- 攻撃例
  - ./a.out 192.2.0.2 20001 www.google.com www.google.com ns.dnslab.jp 216.239.32.10/216.239.34.10/216.239.36.10/216.239.38.10

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃ツールの構造

初期化
socket 2個作成
普通のUDPとRaw
タイマー初期化
初回のクエリ送信

Loop
select
応答を受けたら、注入が成功したか判定
成功したら終了
失敗したら、トリガークエリ送信へ
Tloop時間たったら、トリガークエリ送信へ
Raw socketにかけたら、偽造応答送信へ

終了
成果の表示

トリガークエリ送信
(random)を生成
攻撃対象のport 53へ
クエリを送信

自分のアドレス port *any*
Full-Resolver port 53
ID *any*, Query
(random).トリガードメイン名 A

偽造応答送信
IDをランダムに生成
権威サーバアドレスをランダムに選択
引数であたえたアドレスリストより
偽造応答を生成
Question Sectionはクエリ送信と同じ
Authority Sectionに注入するNS RR
権威サーバのport 53から
攻撃対象の指定ポートへ
Raw socketで送信

権威サーバアドレス port 53
Full-Resolver port *P*
ID *random*, Response
(random).トリガードメイン名 A
注入するNSリソースレコード

Copyright © 2014 Japan Registry Services Co., Ltd.

Tips

- Raw socketでのbyte order
  - FreeBSD: The ip_len and ip_off fields must be provided in host byte order. All other fields must be provided in network byte order.
  - BSD以外はすべてnetwork byte orderのはず
- 終了判定方法
  - 誘導先権威DNSサーバに *.domainname A を書いておくことで、トリガーとなるランダムクエリに存在する応答が戻る
- デバッグツール
  - tcpdump
  - rndc dumpdb

Copyright © 2014 Japan Registry Services Co., Ltd.

実験環境

- ~~箱庭を準備するのは面倒なので~~
- より現実に近い評価にするために
- JPRSの研究ネットワークで実施
  - グローバルアドレスを割り当て
    - DNS trafficをNATしたらNAT箱のCPUがあふれたため
  - ローカルネットワークは1Gbps
  - 対外線は10MbpsでWIDEインターネットに接続
  - 攻撃対象の名前解決クエリが外に漏れるが気にしない
- マシンはVM 2個
  - Xeon 2.5GHz 8コアのHVからcpu 2個, 3GB割り当て
  - 同じネットワークセグメントに設置

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃の前提と実験の条件

- 前提: 攻撃ツール以外からのクエリなし
  - 本物が入ったらTTL時間は攻撃成功しにくいため
- Port randomizationをoffにしたBIND 9, Unboundを用意
  - 203.178.129.2 port 20001
- 誘導先のDNSサーバとしてns.dnslab.jp
  - 実験で攻撃してみたゾーンが多数書いてある
  - *.domainname Aを書いておく (終了判定のため)
- 簡略化のためIPv4でのみ評価

Copyright © 2014 Japan Registry Services Co., Ltd.

実験1: www.google.com (1)

- (random).www.google.com Aクエリを送り
- google.comの権威サーバのIPアドレスから以下のパケットを大量に送信
  - Question: (random).www.google.com A
  - Authority: www.google.com NS ns.dnslab.jp
  - ID ランダム, サーバアドレスを4通りランダムに
- 入るまで(random)を変えながら繰り返し
- 入らないときはrndc dumpdbして確認
  - 正規のAと、偽造NSが混ざることあり

Copyright © 2014 Japan Registry Services Co., Ltd.

実験1: www.google.com (2)

% sudo ./a.out 203.178.129.2 20001 www.google.com www.google.com ns.dnslab.jp 216.239.32.10/216.239.34.10/216.239.36.10/216.239.38.10

elapsed: 46.012747

num_sent_queries=1273　27.666246 pps

num_sent_responses=3070290　66726.944166 pps 2411.853888/queries

num_received=1255　98.586017 %

% drill @203.178.129.2 www.google.com

www.google.com. 86400　IN　A　203.178.129.44

www.google.com. 86139　IN　NS　ns.dnslab.jp.

- 46秒で注入成功、28qpsのクエリと66726ppsの偽造応答

Copyright © 2014 Japan Registry Services Co., Ltd.

実験2: net (1)

- (random).root-servers.net A クエリを送り
- ルートサーバのIPアドレスから以下の応答パケットを大量に送信
  - Question: random.root-servers.net A
  - Authority: net NS ns.dnslab.jp
  - ID, サーバアドレスを変えながら
- 入るまで(random)を変えながら繰り返し
- 入らないときはrndc dumpdbして確認
- 本物のnetがキャッシュされていたらflush

Copyright © 2014 Japan Registry Services Co., Ltd.

実験2: net (2)

% sudo ./a.out 203.178.129.2 20001 root-servers.net net ns.dnslab.jp 198.41.0.4/192.228.79.201/192.33.4.12/199.7.91.13/192.203.230.10/192.5.5.241/192.112.36.4/128.63.2.53/192.36.148.17/192.58.128.30/193.0.14.129/199.7.83.42/202.12.27.33

elapsed: 8.050168

num_sent_queries=241 29.937263 pps

num_sent_responses=556740 69158.805133 pps

% drill @203.178.129.2 www.apnic.net

www.apnic.net. 86400 IN A 203.178.129.44

net. 86272 IN NS ns.dnslab.jp.

- 8秒で注入成功、30qpsのクエリと69158ppsの偽造応答

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃手法のまとめ

- Port Randomizationしていないと数秒から数分で注入できる
- ターゲットはほとんどのドメイン名
  - A, AAAA, PTRなどがついているほとんどのホスト名
  - いくつかの中間のドメイン名 (netなど)
- Port Randomizationしていると、数時間では入らない
  - random.www.google.comやrandom.root-servers.netという怪しいクエリを出し続けるのはいやなので止めた
- 成功しないときは攻撃対象のcacheをflushする
  - net, root-servers.netへの攻撃は、キャッシュにnet, root-servers.net NSがあると成功しない
  - www.google.com Aがあっても、www.google.com NSは入ることに注意

Copyright © 2014 Japan Registry Services Co., Ltd.

試作した手法の欠点

- 攻撃がばれやすい
  - 送ってない応答が大量に到達するので、入出力のパケット数を見るだけでわかる
  - トリガークエリが多めなことと、送っていない応答を捨てる処理のためにフルリゾルバの負荷が上がる
- 攻撃者を特定しやすい
  - トリガーとなるクエリを送り、応答を確認するために、そのIPアドレスを詐称できない
  - 誘導先のDNSサーバを用意する必要がある
- 攻撃側にもリソースが必要
  - 帯域と、高性能機器

Copyright © 2014 Japan Registry Services Co., Ltd.

期待値と実績の比較

	変数	単位	変数の範囲	www.google.com攻撃		net攻撃	
Port randomization				no	yes	no	yes
フルリゾルバポート数	Nport		1~2^16	1	64000	1	64000
ID数	Nqid		2^16	65536	←	65536	←
権威サーバアドレス数	Nns		1~13	4	←	13	←
権威サーバへのRTT	Nauth	秒	0.001~0.2	0.036~0.094	←	0.006~0.272	←
繰り返し時間	Nloop	秒		0.036	←	0.033	←
偽装応答の送出レート	Rans	pps		66726	←	69158	←
一度目攻撃成功確率				0.00916	0.00000014	0.00267	.00000004
攻撃成功の期待値		秒		3.9	251434	12.3	788425
攻撃時間の実測値		秒		46	×	8	×

Copyright © 2014 Japan Registry Services Co., Ltd.

期待値と実績の比較、まとめ

- 期待値と実績の違い
  - www.google.comでは期待値3.9秒、実測値46秒
  - netでは期待値12.3秒、実測値8秒
  - それぞれ一度の試行のため、違いが出た
  - 多数試行して平均を取ると期待値に近づくと考えられる
- Port randomizationしていると注入にかかる時間の期待値がそれぞれ、2.9日と9日になる
  - 3日間も6万ppsの怪しいパケットがくれば気がつくはず

Copyright © 2014 Japan Registry Services Co., Ltd.

攻撃時間を短縮する最適化

$$T = \frac{Nqid * Nport * Nns}{Rans}$$

- 基本的には、偽造応答の送出レートを上げてやることと、$Tloop < Tauth$に注意すること
- たとえば、偽装応答を1Gbpsで送ると応答パケットサイズ100の場合、1,000,000pps
- ポートランダマイゼーションしていない場合

65536 * 1 * 4 / 1000000 = 0.262秒　瞬殺

- ポートランダマイゼーションしている場合

65536 * 64000 * 4 / 1000000 = 16777　4時間半

Copyright © 2014 Japan Registry Services Co., Ltd.

本攻撃手法への対策

- Port Randomization
  - 時間の猶予が得られる
- 監視
- TCP transportの使用
  - 権威サーバとの間の通信をTCPにするとこのツールでは攻撃できない
  - 一般的にはシーケンス番号が32ビットになることと少なくとも2パケットを推定しないといけないため、ハイジャックは困難

Copyright © 2014 Japan Registry Services Co., Ltd.

本攻撃手法への対策: 監視

- フルリゾルバの負荷
  - 多量の不在クエリ
  - 知らない応答
- 入出力のパケット数
  - DNSはクエリ、レスポンスがほぼ一対一なので、非対称なパケット数は攻撃の可能性あり
- 送ってない応答をログに出す
  - http://member.wide.ad.jp/~fujiiwara/ にBIND 9.3.2 patch
  - ログを出すのは重いのでパフォーマンスに注意
- フルリゾルバ権威サーバ間の通信を監視
  - Farsight Security, Nominumによるサービスあり
  - 応答を過去のログと比較

Copyright © 2014 Japan Registry Services Co., Ltd.

DNSSEC検証による攻撃検知

- BIND 9とUnboundではDNSSEC検証していてもキャッシュには入る
- DNSSEC検証によりエラー検知でき、スタブリゾルバにはServer failureを返す
- netにNSを注入できても、net DNSKEYを検証できないため、検証エラー
- TLDは概ねDNSSEC対応
  - ただし複雑な構造を持つTLDでNSEC3 OptOutしている場合に、検証できない中間ドメイン名ができる場合がある
  - RFC 5155を細かく読むとわかるし、対策も簡単
  - co.jpはあの対策では変化しなかった
  - 別のドメイン名をいってもらえたら反応したのに、、、

Copyright © 2014 Japan Registry Services Co., Ltd.

まとめ

- 委任情報を注入する攻撃は容易である
  - ほとんどのドメイン名を攻略できる
  - しかし、足がつく
- Port randomizationで攻撃の効果を弱くできる
  - DNSSEC検証で検知可能
- 守るにはPort randomizationと監視が重要
- 攻撃にはそれなりの体力がいるので、攻撃者に利益があるところしか狙われにくい
- だれかraw socketでのDoSツールに詳しい人いらっしゃいませんか、性能上げたいので

Copyright © 2014 Japan Registry Services Co., Ltd.