SPNセキュリティ技術解説セミナー

DNSにおけるキャッシュ汚染攻撃

2008年10月25日（土）

【2008年11月3日改訂】

塩月 誠人 <mshio@Sec-Pro.Net>

合同会社セキュリティ・プロフェッショナルズ・ネットワーク

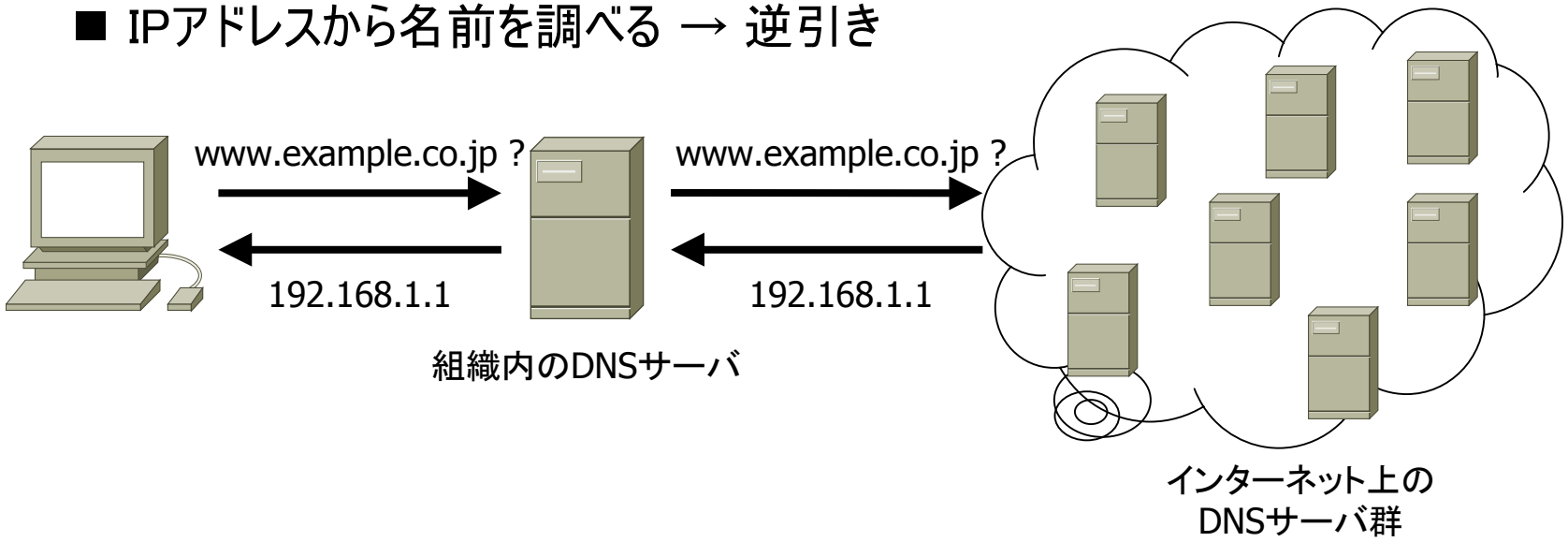
Security Professionals Network Inc.

本セッションの概要

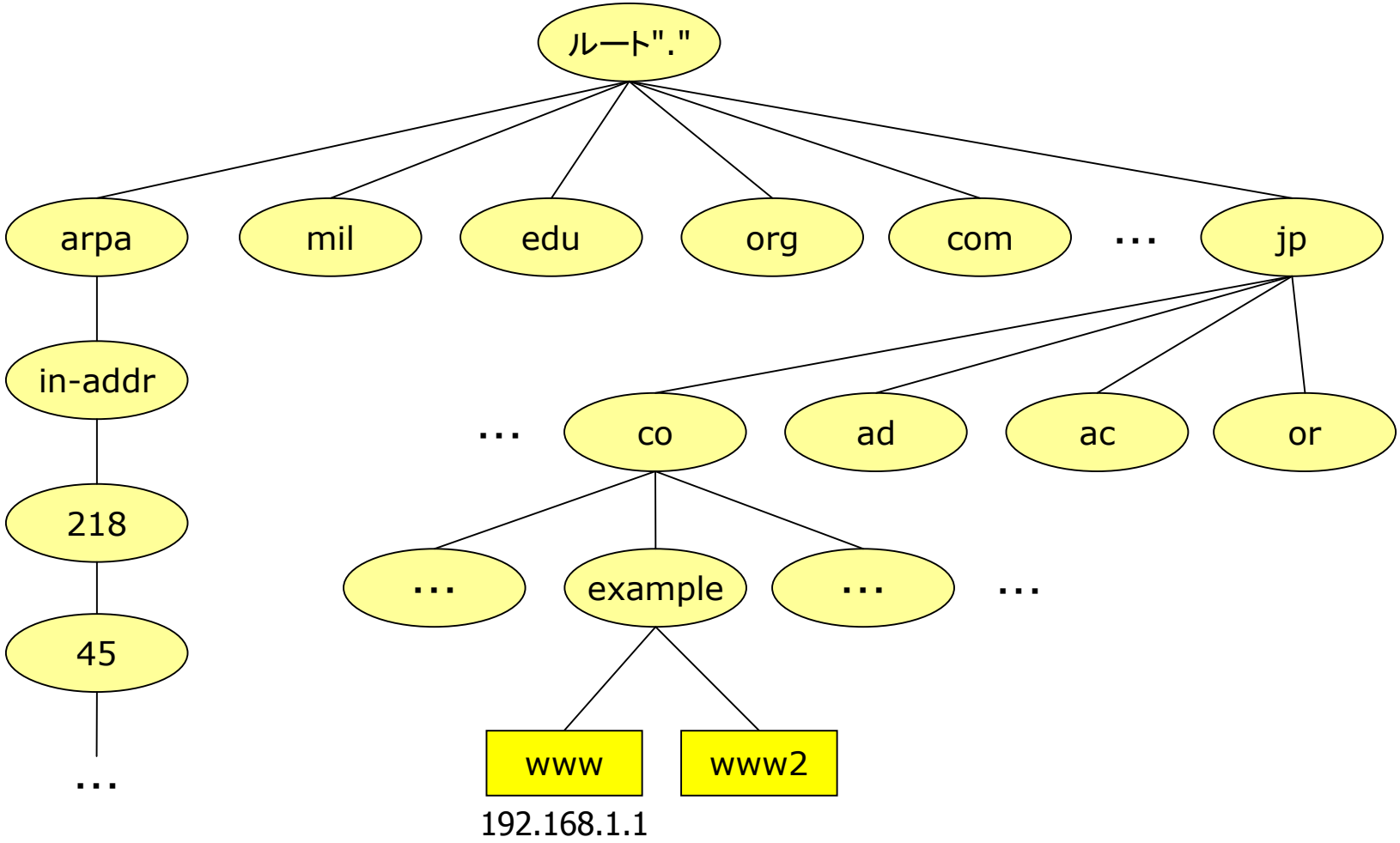
- DNSはインターネットを利用する際に誰もがお世話になる非常に基本的なサービスです。そのためDNSサーバにおけるセキュリティ侵害は、多くのインターネット利用者に重大な影響を与えかねません。中でもDNSキャッシュの汚染は、効果的なファーミング攻撃手法として特に近年注目されています。
- 本セッションではこのDNSキャッシュ汚染が発生する仕組みについて、クエリIDの予測、バースデイアタック、古典的キャッシュ汚染攻撃（Kashpureff Attack）、Kaminsky's Attackといった手法を例に挙げ解説するとともに、その対策方法について考察します。

DNSとは

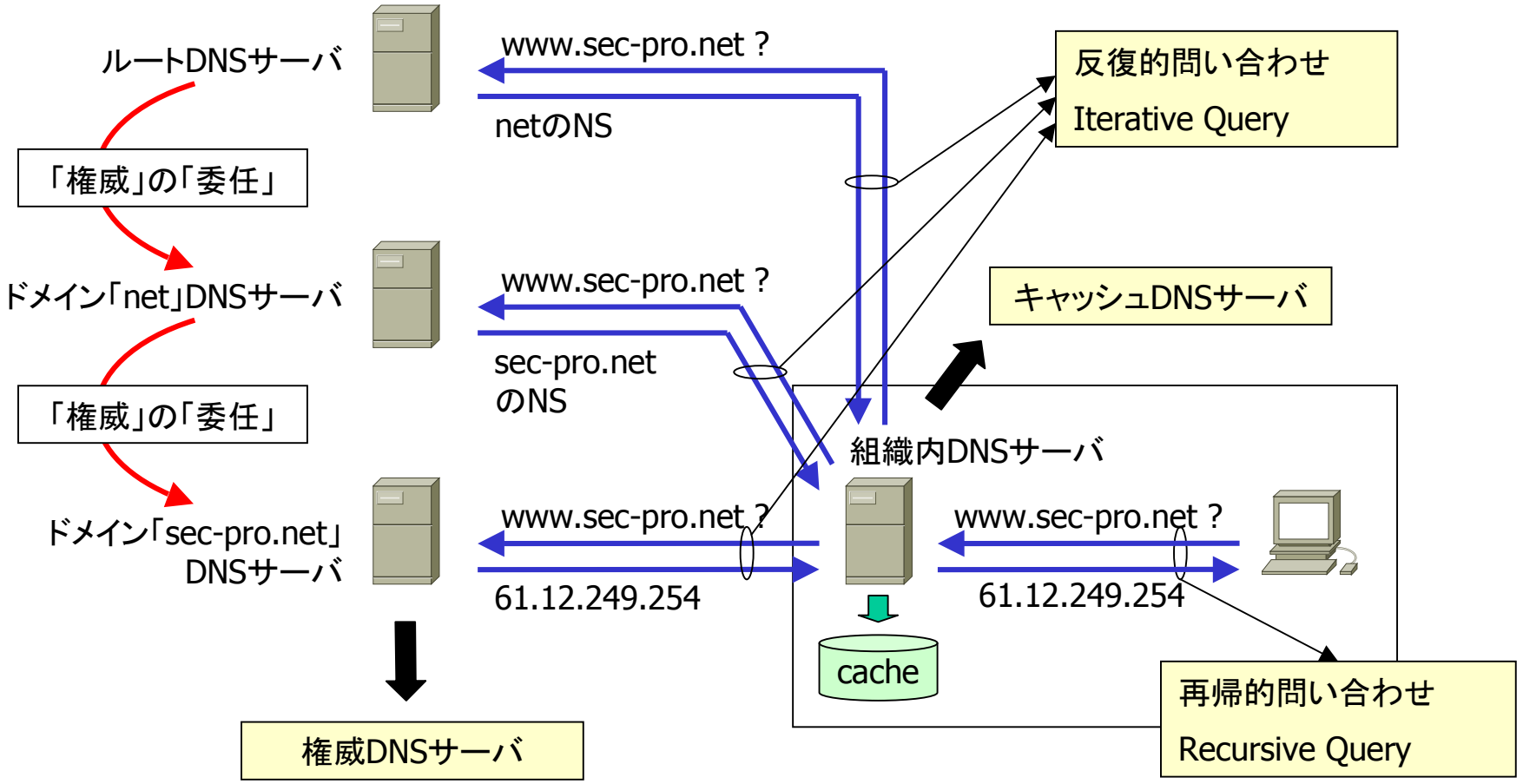
- DNS ・・・ Domain Name System
- 平たく言うと、「ホストの名前とIPアドレスを関連付ける仕組み」
  - 名前の例 ・・・ www.example.co.jp
  - IPアドレスの例 ・・・ 192.168.1.1
  - 名前からIPアドレスを調べる → 正引き
  - IPアドレスから名前を調べる → 逆引き



DNSの階層構造

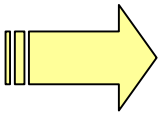

ルート"."
arpa
mil
edu
org
com
…
jp
in-addr
…
co
ad
ac
or
218
…
example
…
…
45
…
www
www2
192.168.1.1

DNSにおける名前解決の仕組み



一般的なDNSの問題点

- 情報の漏洩
  - BINDバージョン情報の取得（BINDに固有の問題）
  - ゾーン転送
- 情報の改竄、不正な情報の挿入
  - ダイナミックアップデート
  - 偽造DNSリプライの挿入
  - DNSキャッシュの汚染



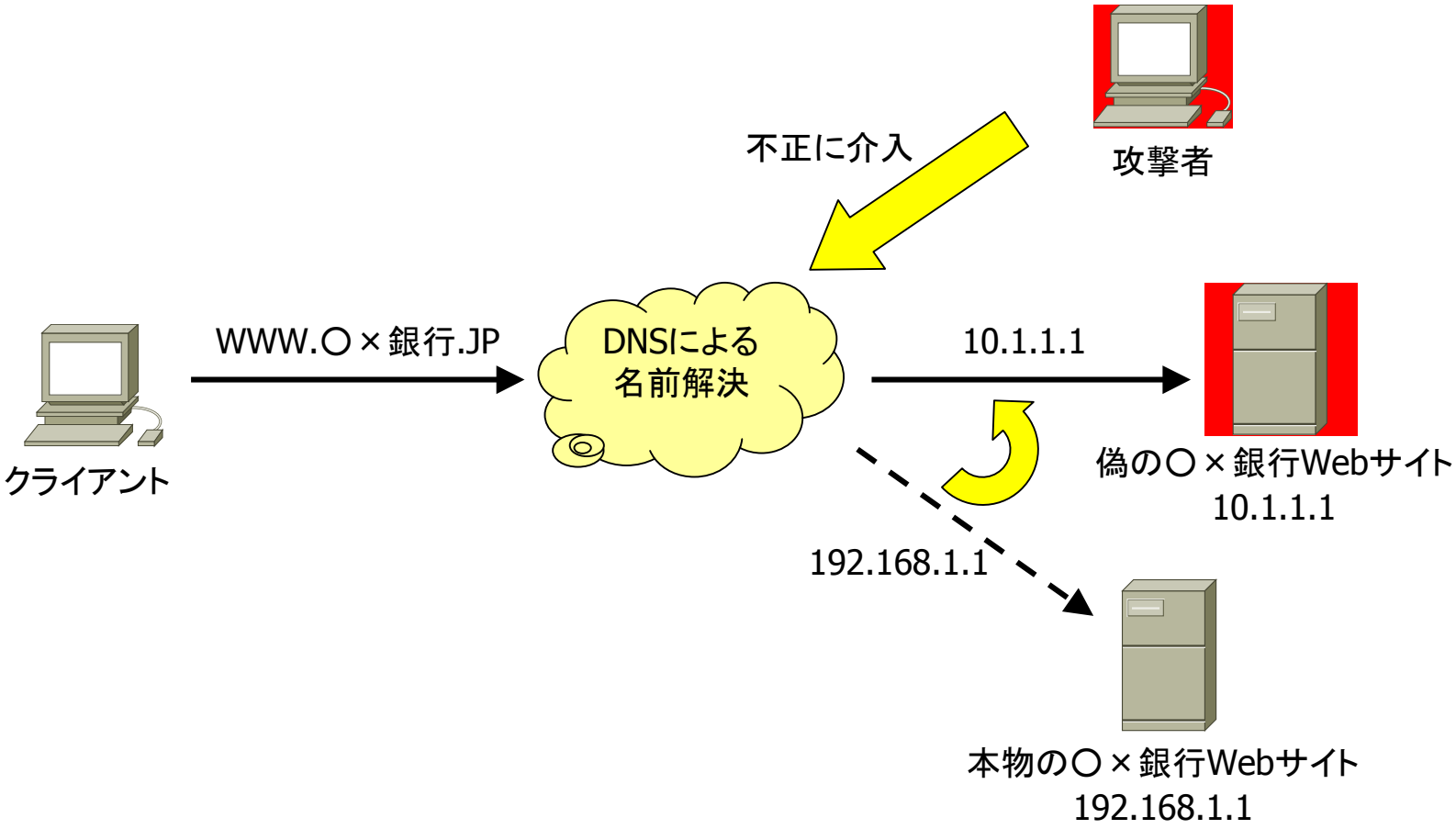
Pharming攻撃に利用

- 外部からの再帰的問い合わせ
  - 資源の浪費（CPU、キャッシュ）
  - 各種DNSサーバ攻撃の助長
- その他、DNSサービスのバグによる各種脆弱性を利用した攻撃（サービス妨害、不正コードの実行、・・・）

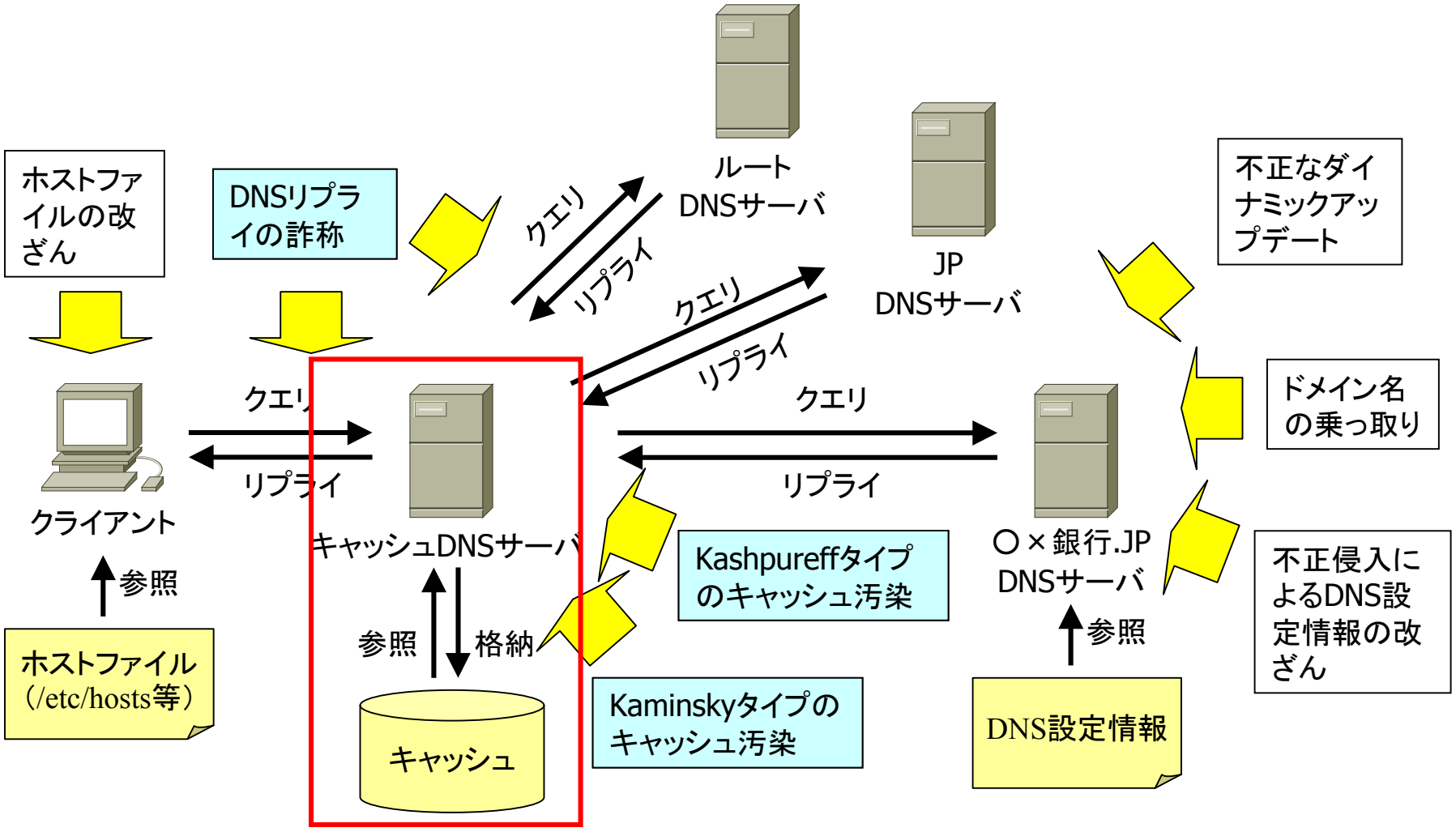
Pharmingの脅威

- Phishing ・・・ エサ（電子メール等）を使って一本釣り
  - Fishing（釣り）から派生した造語
  - 不正な電子メール等を用い、犠牲者を偽サイトに誘導
- Pharming ・・・ タネ（不正な名前解決）をまいて、一気に収穫
  - Farming（農場経営）から派生した造語
  - ホストの名前解決の仕組みを不正に操作し、その仕組みを利用する犠牲者すべてを偽サイトに誘導 → DNSの情報を不正に操作
- いずれも主としてオンライン詐欺（カード番号やパスワード等の不正取得）に利用
- Pharmingの方が効率的かつ効果的
- 「The Pharming Guide」
  - http://www.ngssoftware.com/papers/ThePharmingGuide.pdf

Pharmingの概念



各種Pharming手法



DNSのキャッシュとは

- キャッシュDNSサーバは、問合せた結果のレコードを一定期間保持
  → DNSレコードのキャッシュ
- 同じ問合せに対して、キャッシュに保持されたレコードを返すことでレスポンス時間を短縮、ネットワークトラフィック等の資源を節約
- キャッシュの情報が汚染される（誤った情報がキャッシュされる）と、そのキャッシュDNSサーバに問合せるすべてのホストが騙される
- キャッシュに保持される期間は、当該レコードのTTL値に依存
  - TTL値は権威DNSサーバで定義
  - 通常は1日（86400sec）に設定、最近は短い値の場合も多い
- 攻撃者はキャッシュ汚染時に自由なTTL値を設定することが可能
  - ただし実装によりキャッシュ保持時間の最大値が異なる（?）
    - Windows Server 2003 DNS ・・・ 86400sec (1 day)
    - BIND 9 ・・・ 604800sec (1 week)

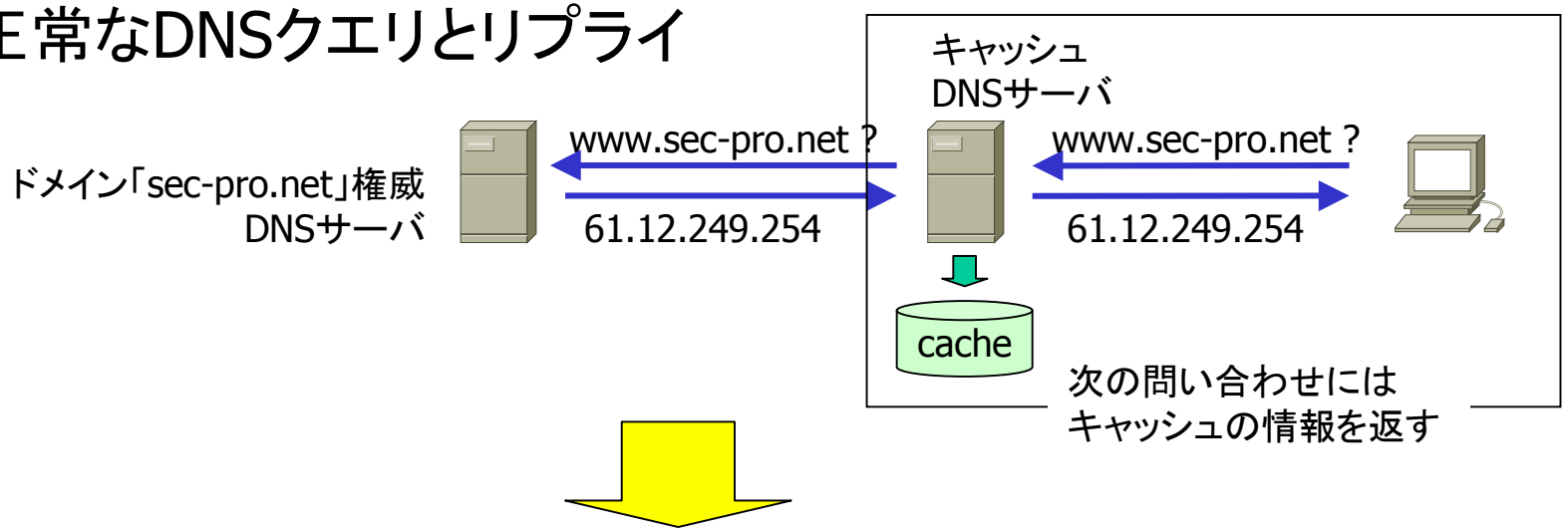
Security Professionals Network Inc.

キャッシュ汚染手法の分類

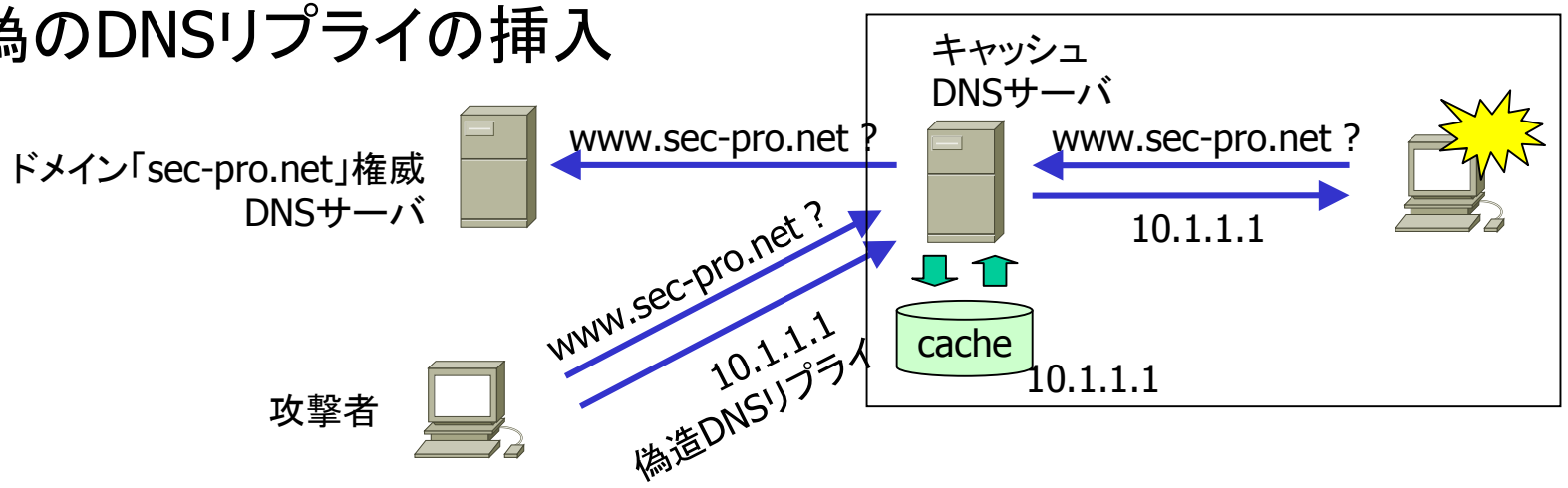
- DNSリプライの詐称によるキャッシュ汚染
  - 偽のリプライを返すことで結果的にキャッシュを汚染
  - いかにしてクエリIDを合致させるかがポイント
    1) MITM/ネットワーク盗聴
    2) ブルートフォース
    3) クエリID予測
    4) バースデイアタック
- Kashpureffタイプのキャッシュ汚染
  - DNSリプライに権威外のレコードを入れて返すことでキャッシュを汚染
  - 古典的キャッシュ汚染手法（1997年、2005年にインターネット上で攻撃）
- Kaminskyタイプのキャッシュ汚染
  - リプライ詐称を効率的に実行し、権威の委任を利用して偽情報を入れる
  - 2008年にKaminsky氏により発表、同7月に各種DNSサーバが一斉パッチ

DNSリプライの詐称によるキャッシュ汚染

正常なDNSクエリとリプライ



偽のDNSリプライの挿入



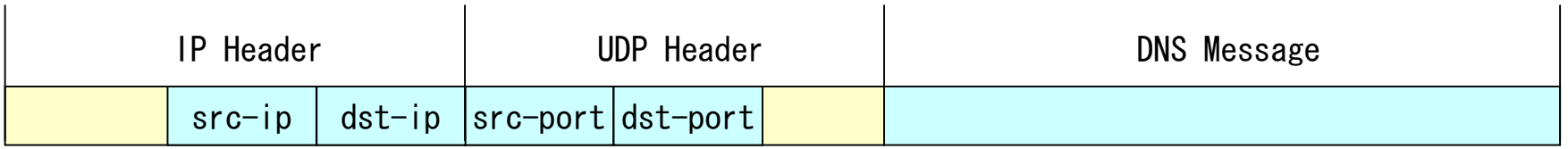
DNSリプライの詐称が成功する条件(1)

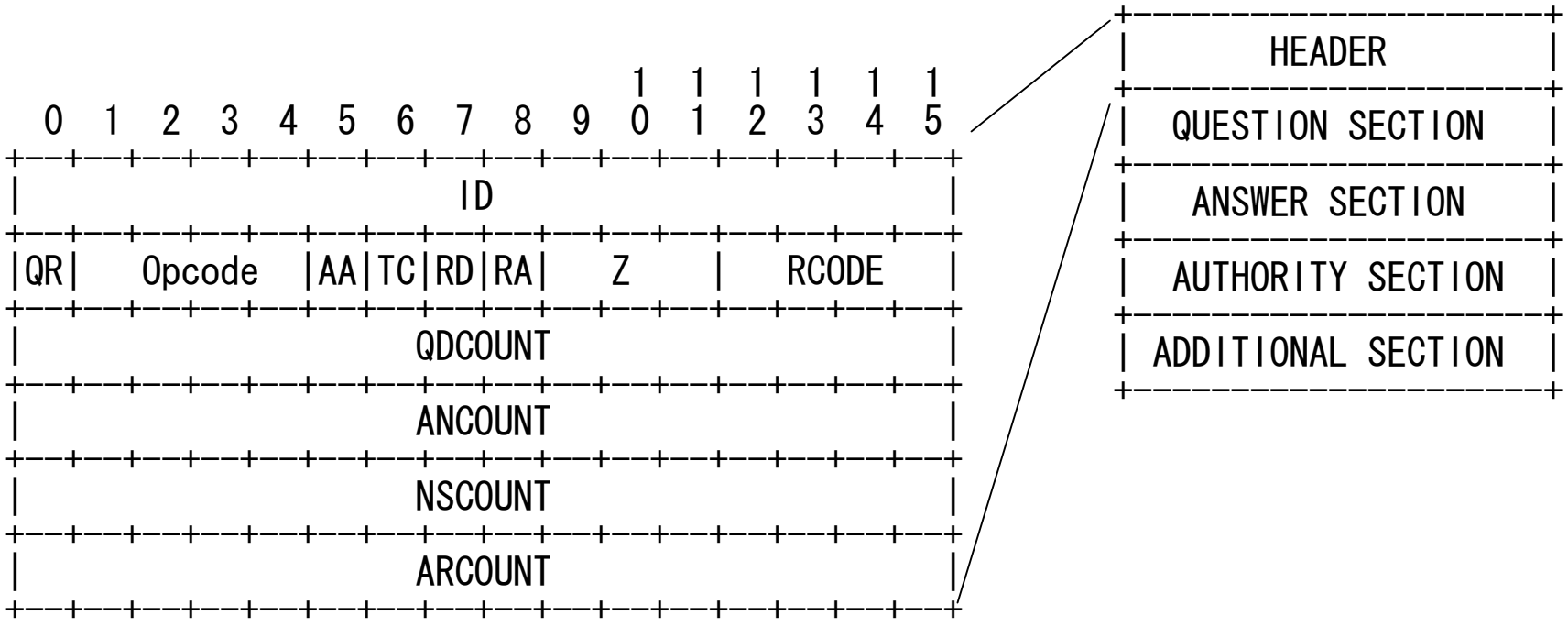
- DNSの応答パケットは基本的にUDPなので、トランスポート層での詐称は容易
- 偽のDNSリプライを挿入するためには以下の条件が必要
  - 正規のDNSサーバからのリプライよりも先に送り込まなければならない
    - 正規のDNSサーバが遠い、動作していない、DOSでダウン、・・・ → 成功率UP
  - 以下が正しいリプライパケットでなければならない
    - ソースおよびデスティネーションIPアドレス
    - ソースおよびデスティネーションUDPポート番号
    - クエリの内容
    - クエリID
- IPアドレス?
  - キャッシュDNSサーバ/権威DNSサーバいずれもIPアドレスは既知

DNSリプライの詐称が成功する条件(2)

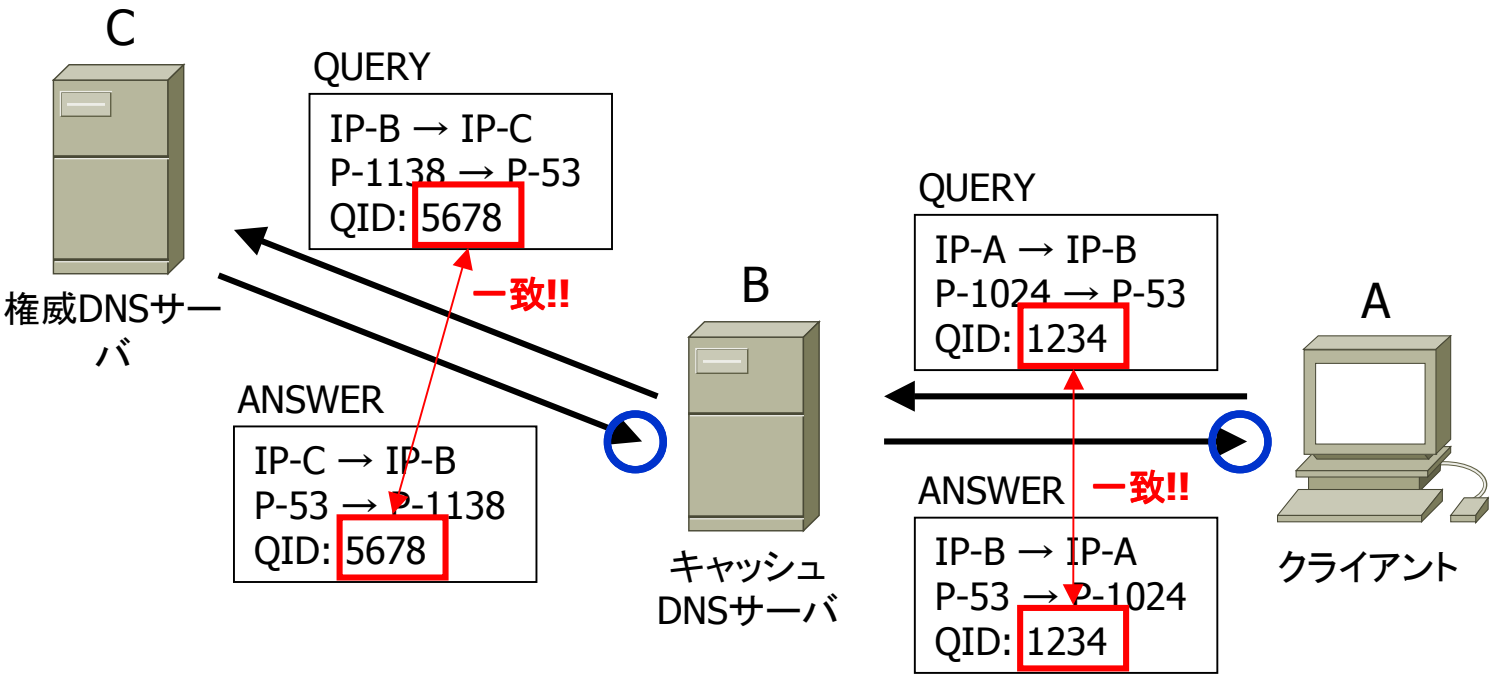
- ポート番号?
  - 権威DNSサーバ側 ・・・ 53番固定
  - キャッシュDNSサーバ側 ・・・ 今年7月の一斉パッチ以前は多くの場合固定のため、事前に問合せることで判明
- クエリの内容?
  - 攻撃者が能動的に問合わせを発行すれば既知
- クエリID(Transaction ID)?
  - DNSの問い合わせの識別子(2バイト → 65,536通り)
  - 古いBINDやNT4.0DNS(SP4より前?)では連番だったため、容易に予測可能であったが、最近のDNSサービスは基本的にランダムなクエリIDを使用
- ローカルネットワーク上では比較的容易に詐称可能(ネットワーク盗聴やMITM攻撃を利用)

DNSリプライパケット

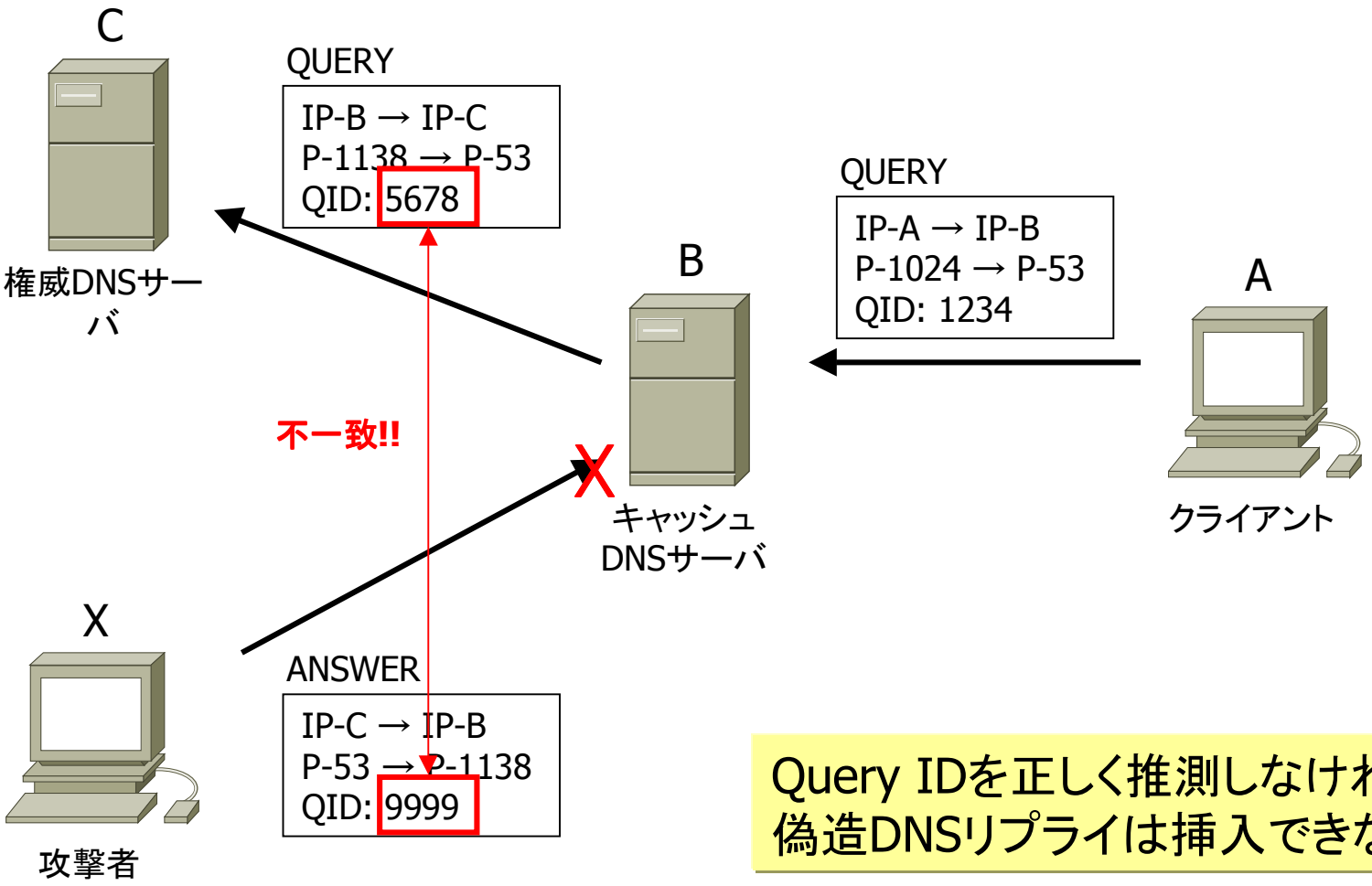




クエリIDによるリプライ詐称の防止(1)



クエリIDによるリプライ詐称の防止(2)



Query IDを正しく推測しなければ
偽造DNSリプライは挿入できない

偽造リプライ挿入の確率

- キャッシュDNSサーバが偽造リプライを受付ける確率は、

$$P_s = \frac{R * W}{N * P * I}$$

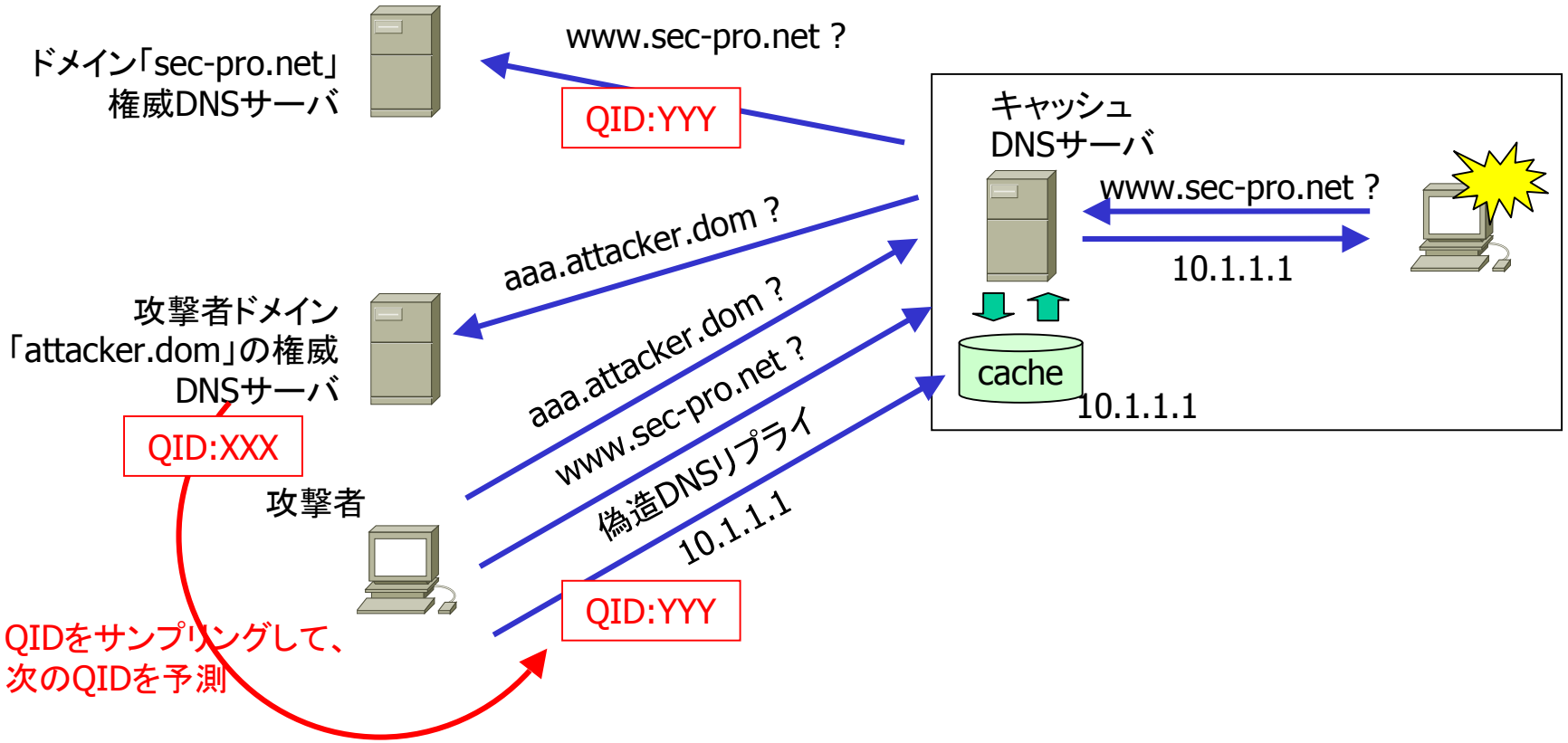
R: 偽造リプライパケットの送出レート（pps）、つまり一秒に何発打てるか
W:「窓」が開いている秒数（s）、つまり正規リプライが返るまでの時間
N: 対象ドメインの権威DNSサーバの数
P: キャッシュサーバのポート番号がとりうる値の数（1～64,000程度）
I: Query IDのとりうる値の数（16bit、つまり「65,536」）

- R=5000, W=0.1, N=2, P=1, I=65536として、0.4%程度の確率

クエリID予測によるDNSリプライの詐称(1)

- 以下のDNSサーバのクエリIDは容易に予測できることが2007年に公開
  - BIND8(8.4.7-P1にてフィックス)
    - http://www.trusteer.com/docs/bind8dns.html
  - BIND9(9.2.8-P1, 9.3.4-P1, 9.4.1-P1, 9.5.0a6にてフィックス)
    - http://www.trusteer.com/docs/bind9dns.html
  - Windows 2000 Server/Windows Server 2003(MS07-062にてフィックス)
    - http://www.trusteer.com/docs/windowsdns.html
- 攻撃者の管理するDNSサーバに問い合わせを発行させ、そのクエリIDを基にして次のクエリIDを予測し、偽造DNSリプライを挿入

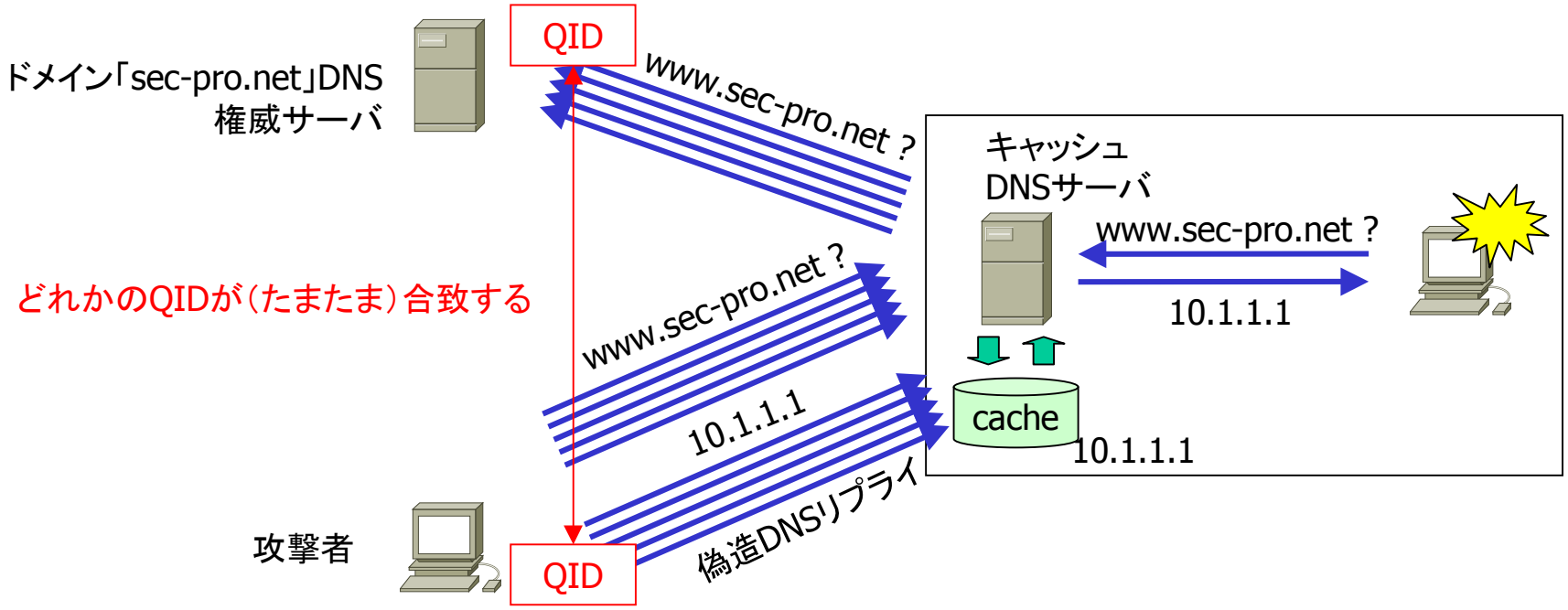
クエリID予測によるDNSリプライの詐称(2)



バースデイアタックによるDNSリプライの詐称(1)

- バースデイパラドックス ・・・「人が23人集まると、同じ誕生日同士の人が存在する可能性は1/2を超え、60人も集まると確実に存在する」という確率論
- つまり、自分と同じ誕生日の人を探すのは大変だが、「誕生日が同じ」というペアを探すのは容易
- DNSのクエリIDの場合、500発程度のクエリおよび偽造リプライを送信することにより、80%以上の確率でクエリIDが合致
- DNS Cache Poisoning – The Next Generation
  - http://www.lurhq.com/dnscache.pdf
- BIND9では同時問い合わせを同時並行的に処理しないので安全
- Windows DNSはMS07-062適用後、同時並行的に処理しない

バースデイアタックによるDNSリプライの詐称(2)



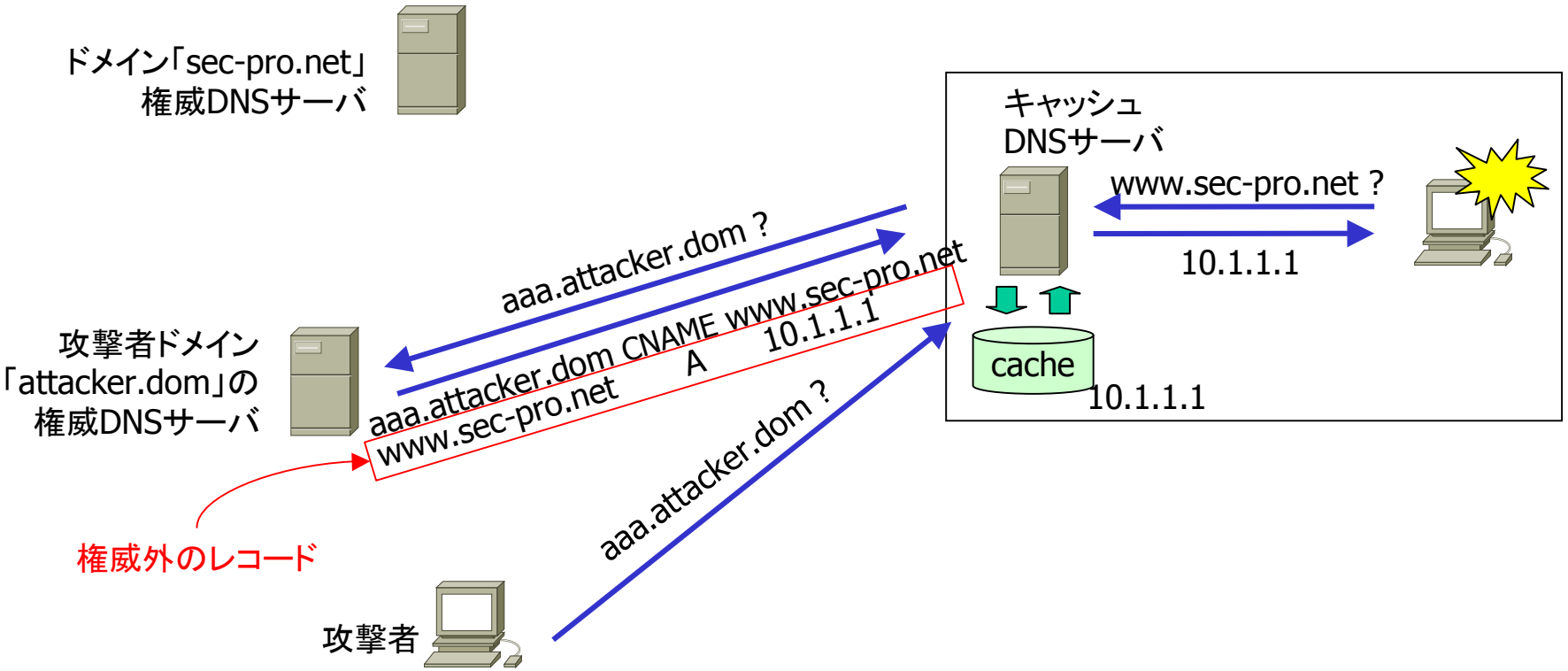
Kashpureffタイプのキャッシュ汚染手法(1)

- キャッシュDNSから攻撃者が管理する権威DNSサーバへ問合わせをさせ、そのリプライに入れた権威外のレコードでキャッシュを汚染
  - CNAMEやNSレコードを利用しANSWER/ADDITIONALセクションに挿入
- 本来信用してはいけないレコードを信用してしまうといった、キャッシュDNSサーバの脆弱性を利用したもの
- 脆弱なDNSサーバは、権威外のレコードをキャッシュに保存
  - 古いBINDや、デフォルトのNT4.0/2000 DNS (SP3より前)は脆弱
  - 最新のWindows DNSでも、設定によっては脆弱になるので注意
  - DNSキャッシュ機能はさまざまな製品(ルータ、ゲートウェイ製品等)に含まれるが、場合によってはそれらに脆弱性が存在
- Windows DNSでの対策:
  - DNS設定の「詳細」プロパティで「Pollutionに対してセキュリティでキャッシュを保護する」を設定(2000 SP3以降および2003ではデフォルト)

Kashpureffタイプのキャッシュ汚染手法(2)

- インターネットで発生したKashpureffタイプのキャッシュ汚染攻撃
  - 1997年7月：
    - http://www.cert.org/advisories/CA-1997-22.html
    - BIND4.9.6/8.1.1より前のバージョンに脆弱性
    - AlterNIC（Eugene Kashpureff氏）により多数のDNSサーバが攻撃
    - www.internic.netへのアクセスがAlterNICのサーバへ誘導
  - 2005年3月～4月：
    - http://isc.sans.org/presentations/dnspoisoning.php
    - Windows NT/2000、Symantecゲートウェイ製品のDNS
    - 各種商用サイトのドメイン名がターゲット
    - 不正サイトにリダイレクト（Web、E-Mail、FTP、IMAP/POP、・・・）
    - 不正サイト上でスパイウェアを配布
    - 対策済みのWindows DNSでもBIND4/8をフォワーダとして使っている場合は脆弱

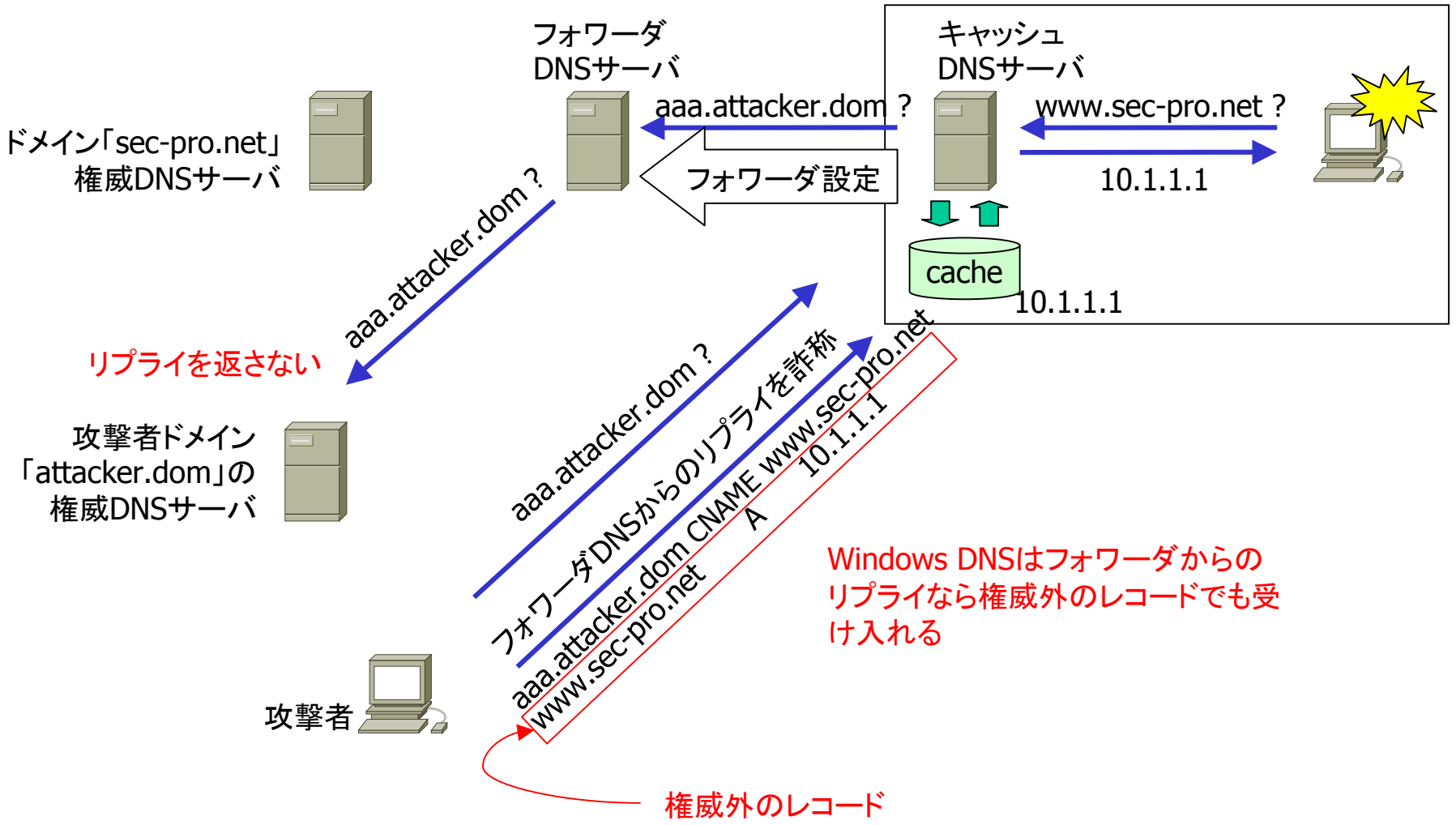
Kashpureffタイプのキャッシュ汚染手法(3)



フォワーダ経由でのキャッシュ汚染(1)

- フォワーダ ・・・ 自分の代わりに名前解決してくれるDNSサーバ
- Windows DNSは、フォワーダの設定をしている場合、フォワーダからのリプライなら権威外レコードであっても受け入れる(仕様)
  - つまり、「Pollutionに対してセキュリティでキャッシュを保護する」と設定していても、Kashpureffタイプのキャッシュ汚染が発生
- 以下の場合、Windows DNSに対するキャッシュ汚染が可能
  - フォワーダDNSにKashpureffタイプのキャッシュ汚染脆弱性がある場合
  - フォワーダDNSが毒を浄化せずにリプライを戻す場合(BIND8/4?)
  - フォワーダDNSを詐称してキャッシュ汚染攻撃された場合
    - クエリIDの推測が必要(バースデイアタックや総当り)
  - 解説ページ: http://www.st.rim.or.jp/~shio/advisories/windns_poison/windns_poison.txt

フォワーダ経由でのキャッシュ汚染(2)



従来のキャッシュ汚染攻撃手法の欠点

- DNSリプライの詐称によるキャッシュ汚染
  - 一度失敗すれば正規のレコードがキャッシュされ、キャッシュ保持中（TTL値の期間中）は攻撃できない ・・・ 効率的な攻撃が不可
    - ある一定の期間（T）連続した攻撃で、一度でも偽造リプライを受付ける確率は、TTLの値に依存

$$P\_cs = 1 - \left(1 - \frac{R * W}{N * P * I}\right)^{T/TTL}$$

    - R=5000, W=0.1, N=2, P=1, I=65536, TTL=86400（1日）として、10日で約4%
    - TTLが例えば1時間（3600）だとすると、10日で約60%、20日で約84%
- Kashpureffタイプのキャッシュ汚染
  - 脆弱性を持つキャッシュDNSサーバのみが攻撃対象 ・・・ 攻撃対象が極めて限定（設定を誤ったWindows DNS等）

Kaminskyタイプのキャッシュ汚染手法(1)

- 2008年になって、Dan Kaminsky氏が新しいタイプのDNSキャッシュ汚染手法を考案
  - DNSのもともとの設計による問題
  - 今まで考えられていたよりも容易にリプライ詐称が可能
  - 攻撃の機会はTTL値に依存しない(いつでも「レース」が開始できる)
  - BIND、Windows DNS等、多くのターゲットに適用可能
  - DNSの下位ドメインに対する「委任」の仕組みを利用してキャッシュを汚染
  - 実装によっては既存レコードの上書きも可能(Windows DNS → Aレコード)
- 7月に各種DNSの実装が一斉にアップデート
  - ソースポートをランダマイズすることで攻撃成功の可能性を低減

Security Professionals Network Inc.

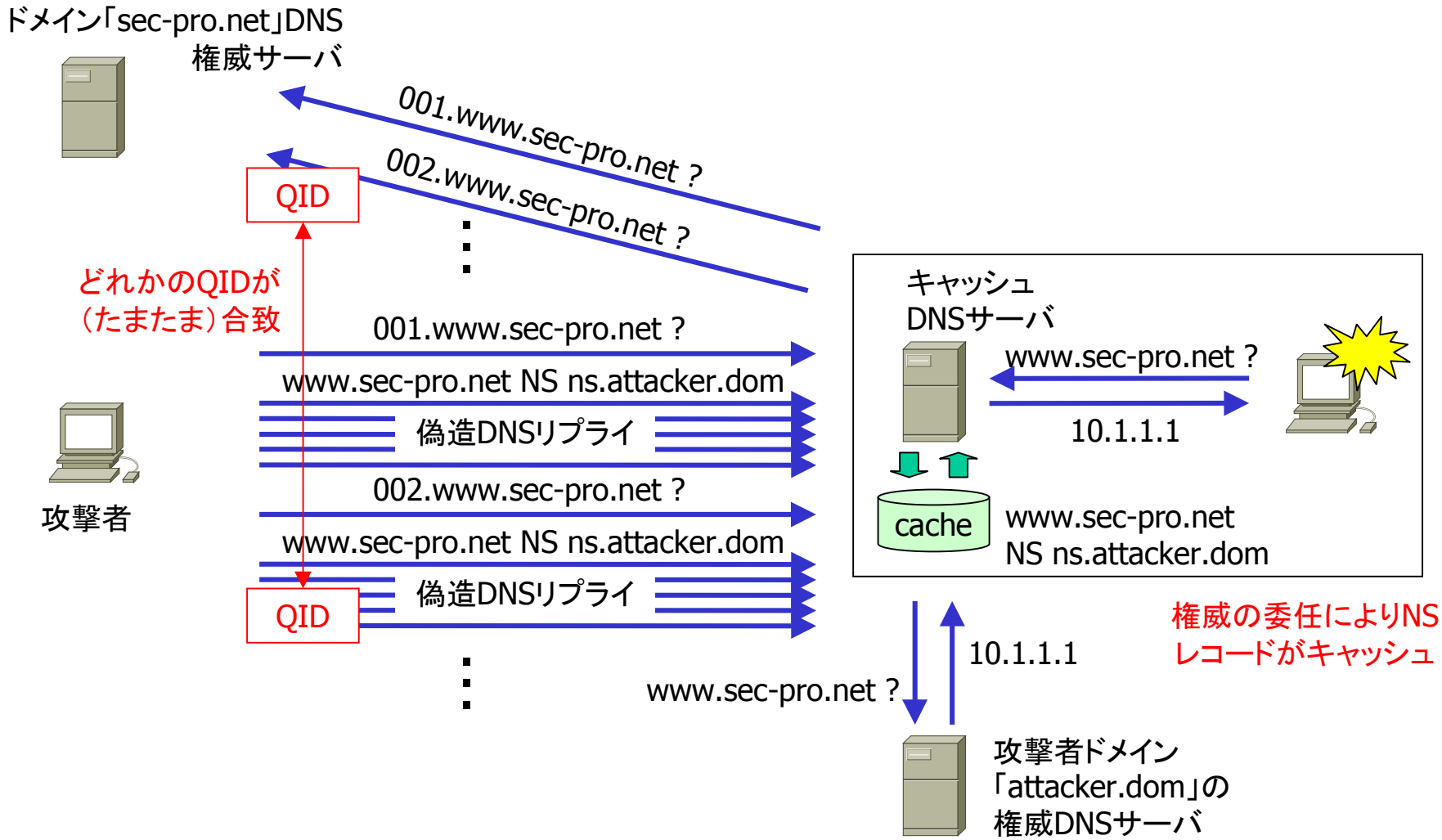
Kaminskyタイプのキャッシュ汚染手法(2)

- 基本はブルートフォース型DNSリプライの詐称
  - つまりターゲットのキャッシュDNSサーバから権威DNSサーバへの問い合わせに対するリプライを詐称
  - ランダムなクエリIDで多数の偽造リプライを返し、どれかが当たるのを待つ
- 存在しないホスト名をクエリに使用
  - 001.www.sec-pro.net等の存在しないレコードを、キャッシュDNSサーバから権威DNSサーバへ問い合わせさせる
  - 毎回クエリのホスト名を変えるため、TTL値に依存することなく何度でも連続してリプライ詐称をトライすることが可能(001, 002, 003, 004, ...)
- どのようにしてキャッシュを汚染するか?
  - 「001.www.sec-pro.net」を偽IPアドレスでキャッシュさせても意味がない
  - やりたいことは「www.sec-pro.net」を偽IPアドレスでキャッシュ

Kaminskyタイプのキャッシュ汚染手法(3)

- 手法その1:
  - 偽造リプライ中に委任先のNSレコードを入れることで参照先を変更
    例1) www.sec-pro.net NS ns.attacker.dom
    例2) sec-pro.net NS ns.attacker.dom
  - それにより、www.sec-pro.netの問い合わせを攻撃者のDNSサーバへ誘導
  - 例2の場合は、sec-pro.netドメイン全体の乗っ取りが可能
  - BIND9.5.0-P2およびWS2K3/SP2(MS08-037適用済み)で検証(いずれもソースポート固定に設定)
- 手法その2:
  - CNAMEやNSレコードを利用して、ターゲットAレコードを直接挿入/上書き
    例) 001.www.sec-pro.net CNAME/NS www.sec-pro.net
    www.sec-pro.net A 10.1.1.1
  - WS2K3 DNSはCNAMEを利用してAレコードの挿入および上書きが可能
  - BIND9.5.0-P2はNSを利用してAレコード挿入が可能

Kaminskyタイプのキャッシュ汚染手法(4)



DNSキャッシュ汚染攻撃への対策(1)

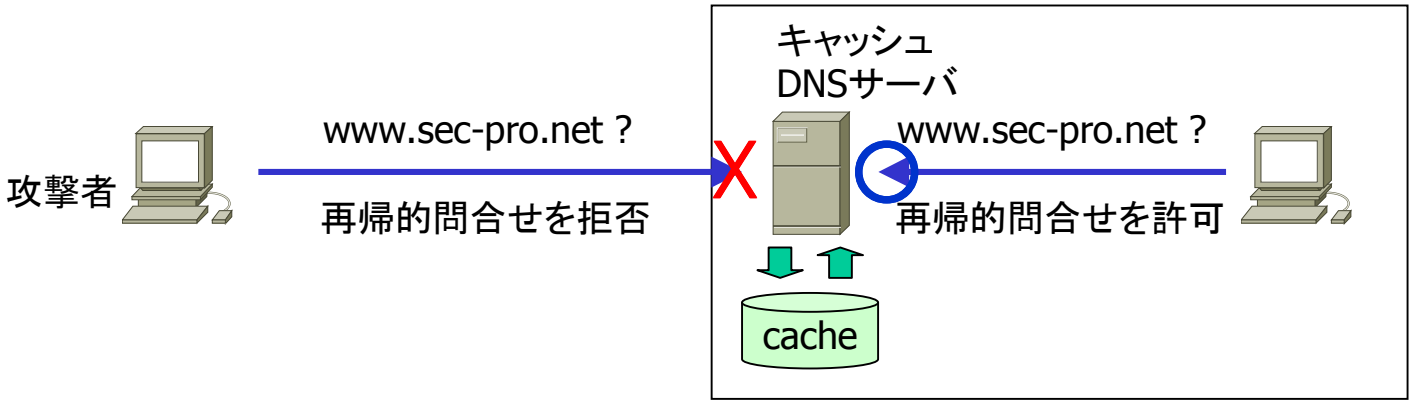
- ソースポートのランダマイズによりキャッシュ汚染成功率を下げる
  - Kaminskyタイプのキャッシュ汚染攻撃が成功する確率

$$P\_cs = 1 - \left(1 - \frac{R * W}{N * P * I}\right)^T$$

    - R=5000, W=0.1, N=2, P=1, I=65536として、1分間の攻撃で約20%
    - ソースポートのランダマイズ → P=64000とすると、10日間の攻撃で約5%
  - 2008年7月に各種DNSの実装が一斉にパッチ提供 → ポートランダマイズ
  - 設定によってはランダム化しない場合もあるので注意
    - 例）BIND ・・・ query-source　port 53; → ソースポートが固定化
  - NAT/NAPT環境ではソースポートのランダム化効果がなくなる場合もある
  - ランダム度のテスト（https://www.dns-oarc.net/oarc/services/dnsentropy）
    - dig +short porttest.dns-oarc.net TXT ・・・ ソースポートのランダム度
    - dig +short txidtest.dns-oarc.net TXT ・・・ クエリIDのランダム度

DNSキャッシュ汚染攻撃への対策(2)

- 外部からの再帰的問合せを禁止する
  - 攻撃者がインターネット経由で直接トリガーを引くことを抑制



  - 内部から強制的に再帰的問合せを発行させることもできるので注意が必要
    - メールサーバに問合せを発行させる(mail from: xxx@www.sec-pro.net等)
    - Webバグを使ってブラウザ/メールクライアントから問合せを発行、・・・
  - 外部からの再帰的問合せのテスト(http://recursive.iana.org/)

DNSキャッシュ汚染攻撃への対策(3)

- 同時並行的な連続問合せの制限(バースデイアタック対策)
- Windows DNSは「Pollutionに対してセキュリティでキャッシュを保護する」を設定(Kashpureff Attack対策)
- パケットフィルタリングによるソースIPアドレス詐称の防止
- DNSSECの導入
  - 公開鍵暗号方式を利用しゾーンのリソースレコードに署名、出所の認証と整合性チェックを提供
  - 問い合わせ側(リゾルバ)はレコードの署名を確認することで、偽造リプライ等によるキャッシュ汚染を回避し、正しいレコードのみを受け入れる
  - 公開鍵を上位ドメインに署名してもうらことで信頼関係を構築
  - インターネットで広く普及するにはしばらく時間がかかりそう
    - SE(スウェーデン):対応中、ORG/GOV:対応予定、・・・
  - DLV(DNSSEC Lookaside Validation)
    - TLDがDNSSEC化していない場合にDNSSEC対応する技術

まとめ

- DNSキャッシュ汚染は極めて影響度の高い攻撃のため、適切な対応が必要
  - DNSサーバの最新版へのアップデート、設定の確認
  - ルータやゲートウェイ製品のDNSキャッシュ機能にも注意
- クエリIDおよびソースポートのランダマイズによりリスクはかなり低減されるが、所詮は確率の大小の問題
- 外部からの再帰的問合せの禁止により攻撃の難易度は高まるが、不可能ではない
- Kashpureffタイプの脆弱性は致命的 → 特にWindows DNSは設定を確認
  - Windows DNSのフォワーダ経由でのキャッシュ汚染にも注意
  - Windowsクライアントでもキャッシュ汚染が発生する可能性
- 最終兵器はDNSSECだが・・・