DNS関連ホットトピックス

2014年1月29日
iijlabセミナー
株式会社日本レジストリサービス（JPRS）
森下 泰宏

本日の進め方

- 「DNS関連ホットトピックス」資料説明（森下：1時間程度）
  - 途中での質問・コメントなどは随時受け付けます
- 内容に関する質疑応答・議論（参加者・藤原・森下）

講師自己紹介

- 氏名: 森下 泰宏(もりした やすひろ)
  - 勤務先:株式会社日本レジストリサービス
  - 肩書: 技術広報担当
- 最近の願いごと:平穏無事な7月
  - 昨年も実現しませんでした・・・(6年連続6回目)
    - 2008年7月:カミンスキー型攻撃手法公開
    - 2009年7月:「BINDコロリ」(パケット一発でBIND 9死亡、回避手段なし)
    - 2010年7月:ルートゾーン署名直後、BIND 9のDNSSEC実装に致命的なバグ発覚(権威DNSサーバーに全力でパケットを送り続ける)
    - 2011年7月:「BINDコロリ」パート2(パケット一発で(以下同文))
    - 2012年7月:BIND 9とNSD 3にそれぞれ2件ずつ、Androidのリゾルバーにキャッシュポイズニング可能な脆弱性発覚(対象:約3億台)
    - 2013年7月:7月最終週の土曜日早朝にBIND 9のゼロデイ脆弱性発表

本日の内容

- DNS Response Rate Limiting（DNS RRL）の概要と現状
- 第一フラグメント便乗攻撃（1st-fragment Piggybacking Attacks）の概要と対策

DNS Response Rate Limiting（DNS RRL）の概要と現状

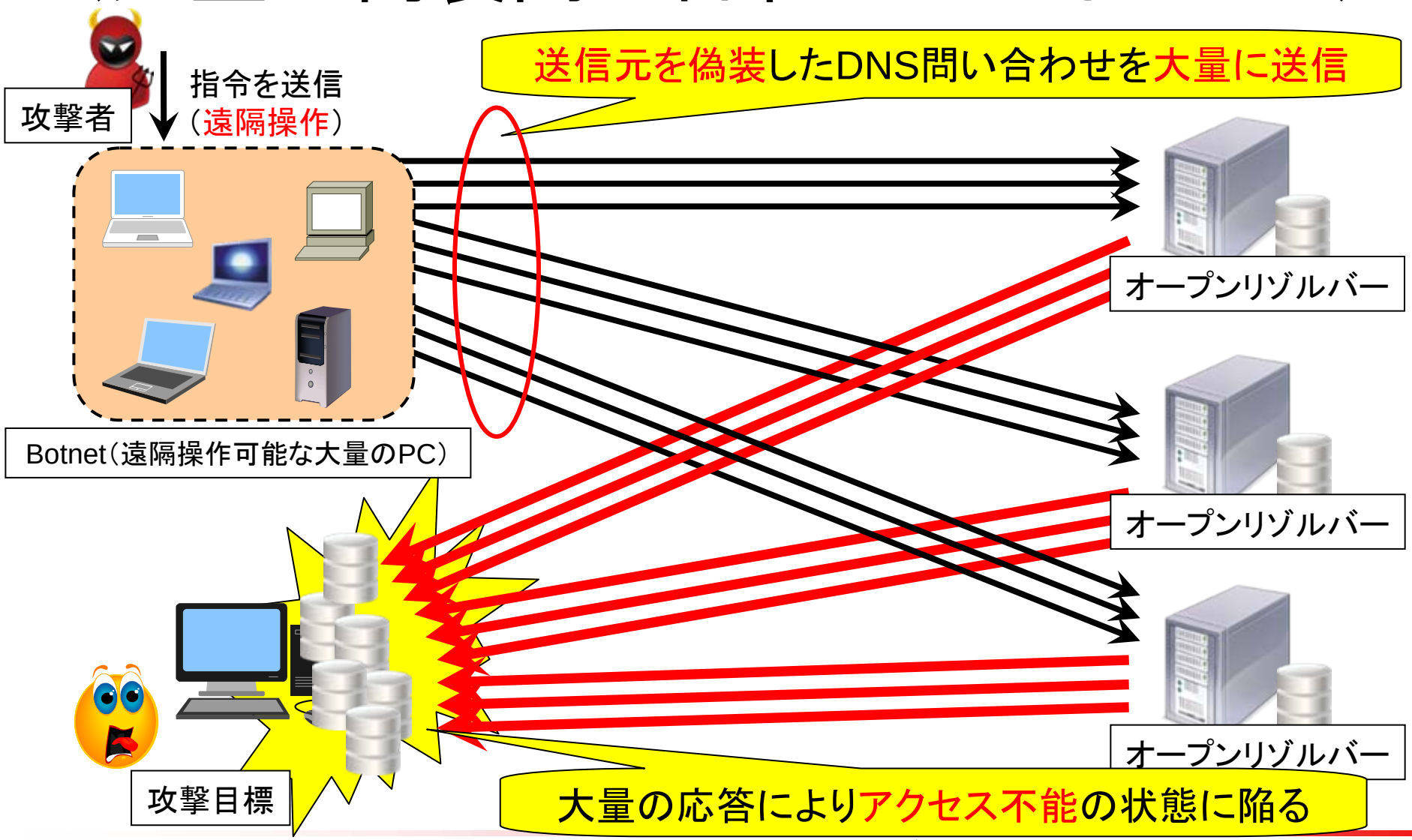
Rate Limiting（レート制限）とは

- 単位あたりにおける何らかの数を制限
  - 単位として時間（秒など）が使われることが多い
- Web技術の分野では従来から一般的
  - 単位時間あたりのAPI実行可能回数
  - 単位時間あたりのダウンロード可能回数
  - 1IPアドレスあたりの同時接続可能数、など
- DNS Response Rate Limiting（DNS RRL）
  - DNSの応答レートを制限するための技術
  - DNSリフレクター攻撃対策として注目されている

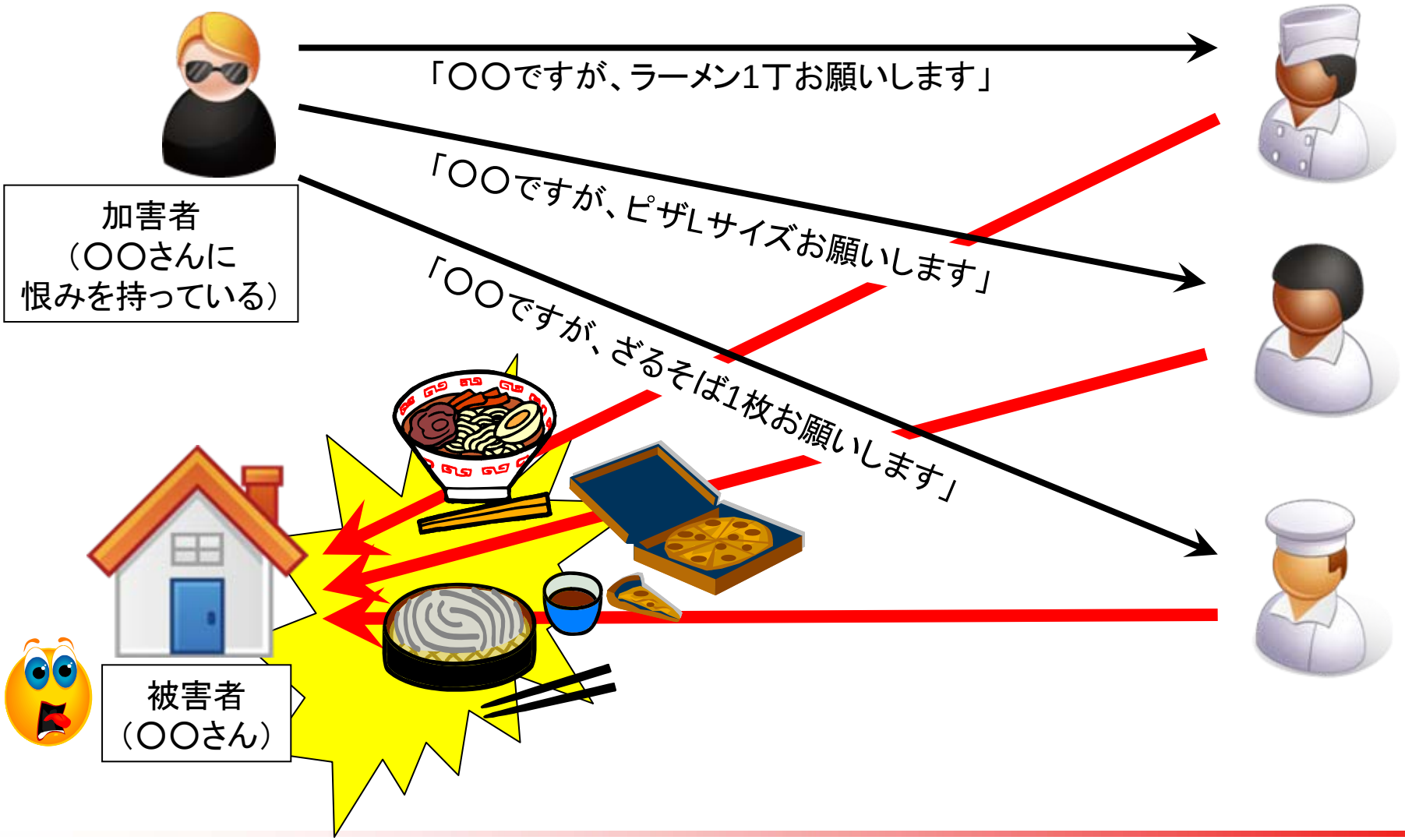
おさらい：DNSリフレクター攻撃

- DNSの持つ特性を利用した攻撃手法
- 別名：DNSリフレクション攻撃、DNS Amp攻撃
  - JPRSではRFC 5358の表記に従い、2013年4月から「DNSリフレクター攻撃」と呼称
    - RFC 5358: Preventing Use of Recursive Nameservers in Reflector Attacks
      <http://www.ietf.org/rfc/rfc5358.txt>
- 攻撃者が送信元（＝応答先）を偽装した問い合わせをDNSサーバーに送信し、DNS応答を攻撃目標に送り付けるように仕向ける

DNSリフレクター攻撃の例（大量の偽装問い合わせによるDDoS）



実社会における類似の事例（なりすまし注文による迷惑行為）



おさらい:オープンリゾルバー

- インターネット上のどこからの再帰検索要求であっても受け付け、処理してしまうDNSサーバー
  - 再帰検索要求:DNSクライアントからの名前解決要求
- インターネット上に多数存在
  - 約2,700万台(2014年1月現在:openresolverproject.org調べ)
- DDoS攻撃の踏み台として悪用
  - 2013年3月には300Gbpsを超える攻撃が発生
- 出自にはいくつかの種類がある
  - 適切なアクセスコントロールがされていないキャッシュDNSサーバー
  - 適切な機能制限がされていない権威DNSサーバー
  - WAN側からの問い合わせも処理してしまうホームルーター

おさらい：オープンリゾルバー対策

- キャッシュ・権威DNSサーバーを機能分離し、その上で以下を実施
  - キャッシュDNSサーバーにおける適切なアクセスコントロール
  - 権威DNSサーバーにおける適切な機能制限
- ホームルーターのファームウェアやハードウェアの更新
- 少しずつではあるが、対策が進みつつある
  - “name and shame” approach（命名：Geoff Huston氏）

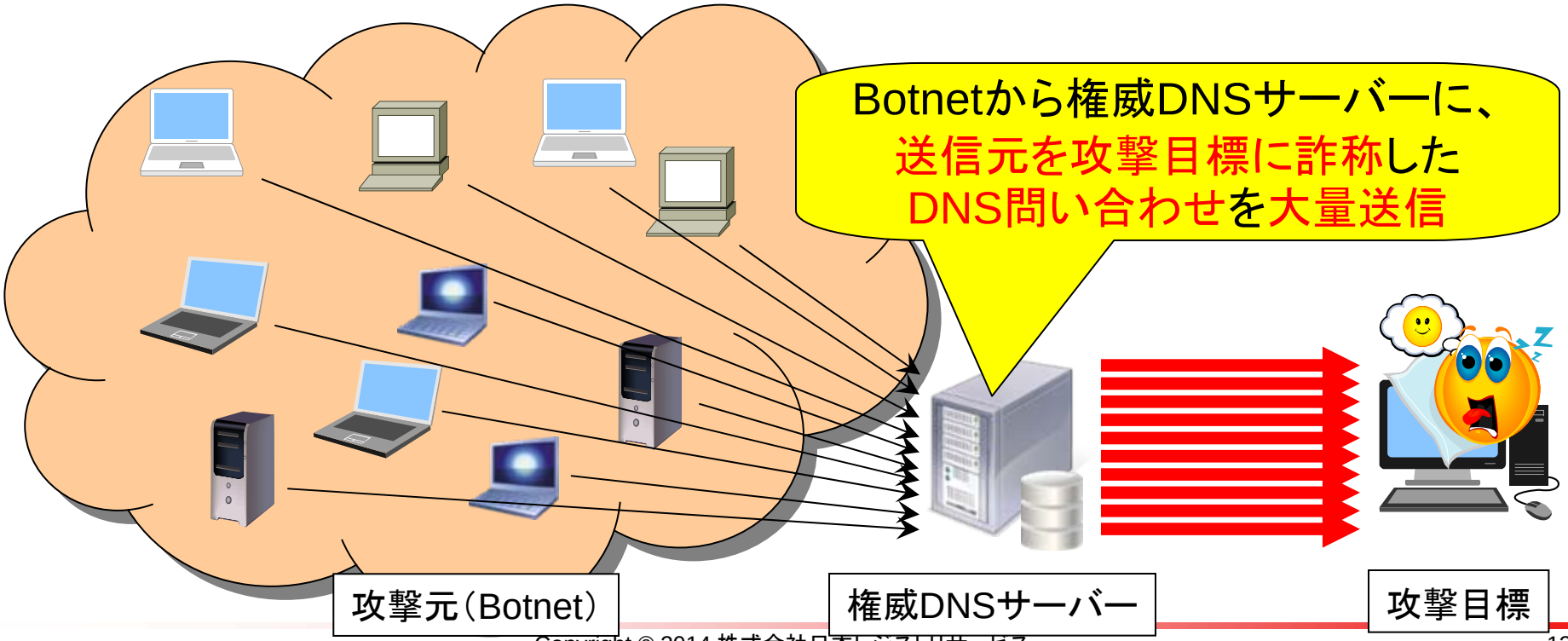
    A Question of DNS Protocols
    <http://www.circleid.com/posts/20130820_a_question_of_dns_protocols/>
  - Telecom-ISAC協議会の場における業界全体での取り組み
  - ISPやレンタルサーバー事業者における地道な対策、など

もう一つのDNSリフレクター攻撃

- 最近、オープンリゾルバーを用いたものに加え、権威DNSサーバーを用いたDNSリフレクター攻撃も観測され始めている
  - 2012年頃から、TLDの権威DNSサーバーなどを利用したDNSリフレクター攻撃の事例が報告され始めた
- 権威DNSサーバーの場合、不適切な設定のDNSサーバーでなくても、DNSリフレクター攻撃に利用可能
  - オープンリゾルバーの場合と異なる

権威DNSサーバーを用いたDNSリフレクター攻撃（1/2）

- 権威DNSサーバーをリフレクター攻撃に直接利用
  - 攻撃用データをキャッシュ（コピー）しないため、攻撃の効率はオープンリゾルバーの場合よりも下がる



権威DNSサーバーを用いたDNSリフレクター攻撃(2/2)

- DNSSEC、IPv6、SPF/DKIM、DANE(RFC 6698)への対応など、DNSの応答サイズそのものが大きくなる傾向にある
- ルートサーバーやTLDの権威DNSサーバーは、IP Anycast、広帯域回線、複数のトランジットなどの施策により、処理能力の強化が図られている
  - つまり、強力なリフレクターとなりうる
- 何らかの有効な対策を実施する必要がある
  - 権威DNSサーバーの特性に応じた対策が必要

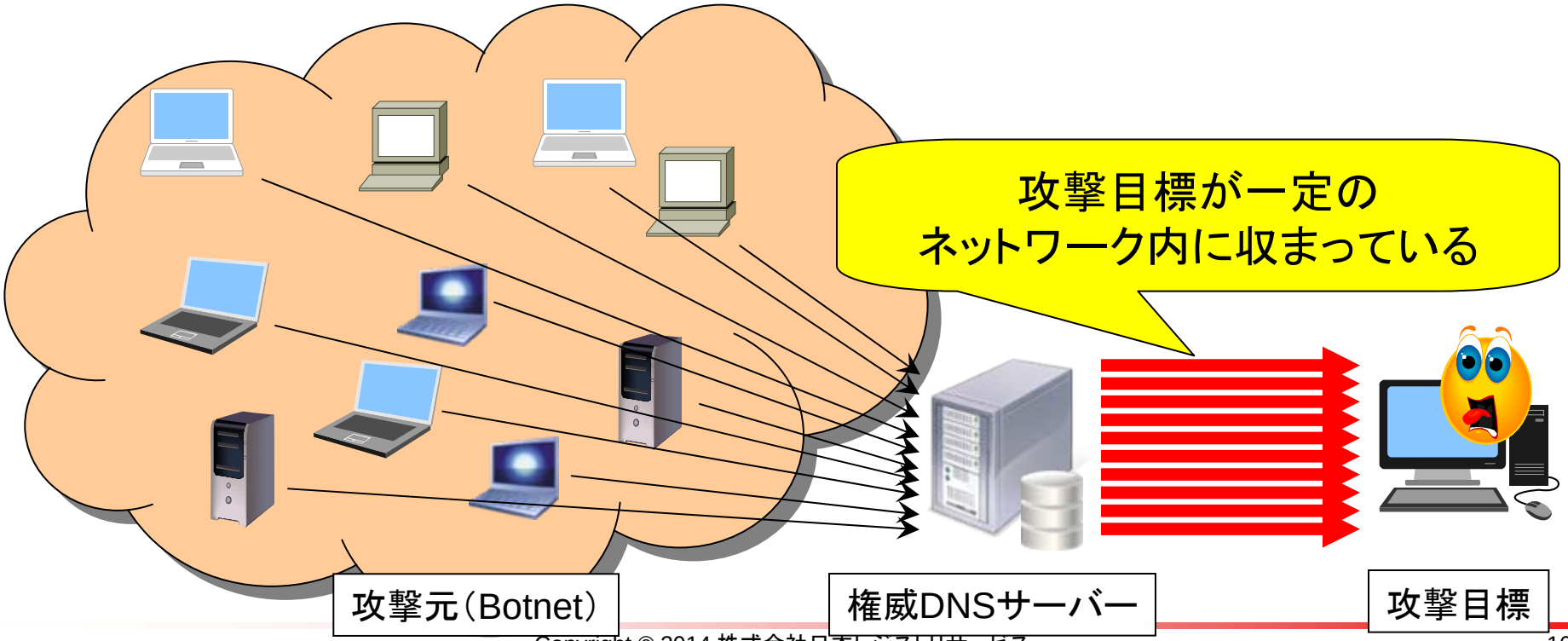
権威DNSサーバーの特性とDNS RRL

- 権威DNSサーバーのサービス対象はインターネット全体
  - キャッシュDNSサーバーの場合、組織/ISP内のみ
- そのため、アクセス制限による事前対策は不可能
  - キャッシュDNSサーバーではアクセス制限が可能
- キャッシュDNSサーバー（オープンリゾルバー）の場合とは別の対策方法を考慮・実施する必要がある

DNS Response Rate Limiting（DNS RRL）は、
そのための有力な対策の一つ

DNS RRLの仕組み（1/3）

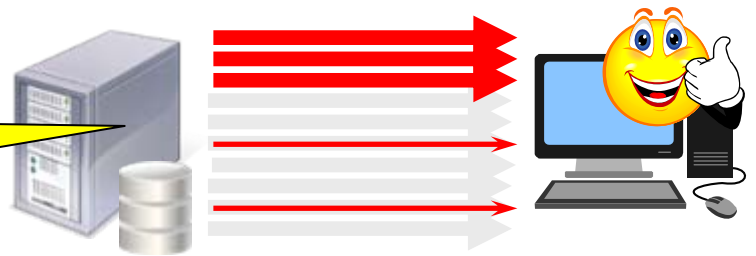
- 攻撃目標（となるIPアドレス）が一定のネットワーク（IPアドレスブロック）内に収まっている点に着目



DNS RRLの仕組み(2/3)

- 事前に決めたルールにより、応答レートを制限
- 具体的には、
  - あるIPアドレスブロック宛の、
  - 単位時間あたりの同一名に対する同一ステータスの応答が所定の頻度を超えた場合に、
- 所定の制限を発動させる
  - 応答の破棄、切り詰め(TC=1)など(詳細は後述)

応答頻度をチェックし、
一定の割合を超えたら応答を制限



DNS RRLの仕組み(3/3)

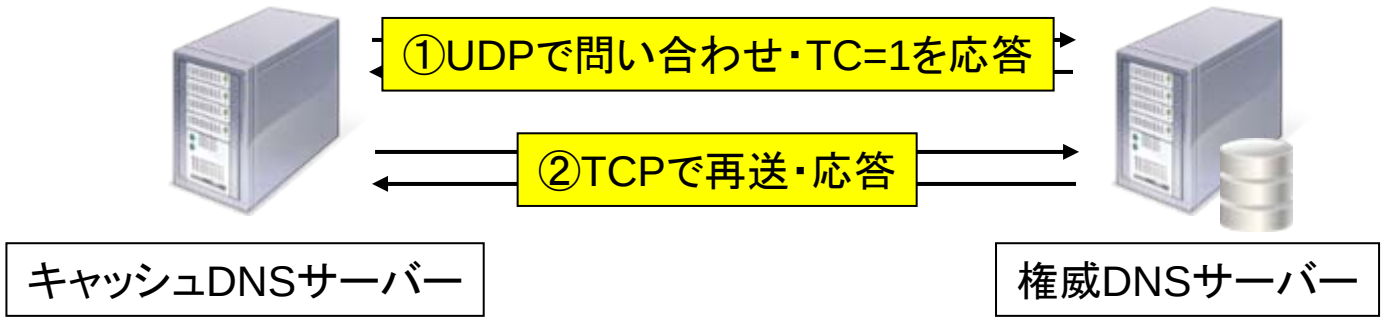
- オープンリゾルバーにおける対策と異なり、DNS問い合わせは制限しない
  - 攻撃の発生そのものの防止ではなく、攻撃の影響の低減が目的
- 名前を変えながらの連続攻撃に対応するため、管理下のサブドメインの不在応答(NXDOMAIN)はすべて、同一名に対する応答と判定される
  - 例:jpゾーンを管理するJP DNSの場合
    - example1.jp、example2.jp、example3.jpのAレコードに対する問い合わせに対する応答(NXDOMAIN)は、すべて同一名に対する応答と判定される

False Positive発生の抑制（1/3）

- False Positive：偽陽性
  - 本来検出すべきでない事象を誤検出してしまうこと
- DNS RRLは検出型の攻撃対策
  - 攻撃があったことを検出し、対策を発動する
  - 検出型の仕組みでは、False Positiveの発生をどのように抑制するかがポイントとなる
- DNS RRLでは、DNSプロトコルの仕組みを利用することでこの問題への対応を図っている
  - TCPフォールバックの仕組みを利用

False Positive発生の抑制(2/3)

- 該当する応答の一部について応答の破棄に替え、DNS応答の切り詰めが発生(TC=1)した旨の、DNSの仕様に合致した応答を返す
  - この動作をslipと呼ぶ
- この応答を、正当な問い合わせを送信したキャッシュDNSサーバーに受信させることでTCPでの再送を促し、正常に名前解決させることを期待している



False Positive発生の抑制(3/3)

- slip設定では、サーバーの処理コスト・ネットワークの負荷・対False Positiveの観点でのトレードオフを考察する必要がある
- BIND 9のDNS RRLのデフォルトでは、該当する応答2回に1回の割合でslip応答を返す(slip:1/2)
  - slipの比率(分母を指定)は設定で変更可能

	応答の破棄	TC=1(slip)応答
サーバーの処理コスト	低	高
ネットワークの負荷	低	中
対False Positive	対応不可	対応可

DNS RRLが効果を発揮する状況

- 権威DNSサーバーにおいて特に効果を発揮
  - 権威DNSサーバーへの問い合わせはキャッシュDNSサーバーからである（はず）
  - キャッシュDNSサーバーにはキャッシュがある（はず）
    - TTL時間内は同一名／タイプの問い合わせを再送信しない（はず）
- キャッシュDNSサーバーへのDNS RRLの安易な導入は、サービスの提供に悪影響を及ぼす危険性あり
  - 導入できないわけではない
  - Google Public DNSでは、独自に実装したRRLを導入
    - <https://developers.google.com/speed/public-dns/docs/security?hl=ja#rate_limit>
- DNS RRLの実装のためにも、キャッシュDNSサーバーと権威DNSサーバーの分離が重要

DNS RRLの実装状況

- 権威DNSサーバーを中心に実装が進んでいる
  - BIND 9
    - 9.9.4以降においてRRLを標準実装
    - それ以前のバージョンに対するパッチも提供
  - NSD
    - 3.2.15以降においてRRLを標準実装
      - デフォルトでは無効、コンパイル時に`--enable-ratelimit`を指定
  - Knot DNS
    - 1.2.0-rc3以降においてRRLを標準実装

DNS RRLの導入状況

- JP DNSサーバー
  - 2013年11月以降に、DNS RRLを導入済
  - ただし、セキュリティ上の理由により具体的な設定内容（設定値など）は非公開
- JP DNSサーバーのほか、いくつかの著名な権威DNSサーバーにおいて導入済（かつ、効果を発揮）・導入作業中である旨報告あり

DNS RRLの注意点（1/2）

- 実運用する際には運用ノウハウの蓄積や、各パラメーターのチューニング・リファインが必要
  - ドキュメントにもその旨の記述あり
  - ログオンリーモードで動作させ、各サーバーの状況に合わせてパラメーターをチューニングすることが有効
- JP DNSサーバーにおける導入事例は、Internet Week 2013 DNS DAYで発表済

DNS RRLの注意点(2/2)

- 「いたちごっこ(cat-and-mouse game)」の技術
  - 攻撃者がより洗練された形(DNS RRLをかいくぐる)に攻撃方法を改良してくることが予想される
- DNS RRLの導入により、DNSキャッシュポイズニングをしやすくなる危険性を指摘する声がある
  - 2013年9月 : フランスCERTAによる指摘
    - Vulnerabilite dans DNS Response Rate Limiting
      <http://www.certa.ssi.gouv.fr/site/CERTA-2013-AVI-506/index.html>
    - CERTAではslip=1(毎回slip)を回避策として提示

まとめ：DNS RRLの特徴

- 権威DNSサーバーにおいて特に効果を発揮
  - キャッシュDNSサーバーへの安易な導入は、サービスの提供に悪影響を及ぼす危険性あり
- 有力かつ巧妙な仕組みであると言える
  - DNSの動作の細部に至るまで緻密に考慮されている
- ただし、実運用にはノウハウの蓄積や、状況に応じた各パラメーターのチューニング・リファインが必要
  - 枯れている技術であるとは（まだ）言えない
- 「いたちごっこ」の技術であり、RRLを回避する形での攻撃方法の洗練が予想される

参考リンク（DNS RRL関連技術資料）

- JPRSの技術解説
  - 「DNS Reflector Attacks（DNSリフレクター攻撃）」について
    <http://jprs.jp/tech/notice/2013-04-18-reflector-attacks.html>
- DNS RRLの技術仕様
  - DNS Response Rate Limiting (DNS RRL)
    <http://ss.vix.su/~vixie/isc-tn-2012-1.txt>
- W. Matthijs Mekking氏（NLnet Labs）の発表資料
  - DNS Rate Limiting
    <http://www.guug.de/veranstaltungen/ffg2013/talks/DNS_Rate_Limiting__Matthijs_Mekking.pdf>
- 山口崇徳氏（IIJ）の発表資料
  - DNS Response Rate Limiting (DNS RRL)
    <http://www.dnsops.jp/event/20130529/dnssec2013springforum-yamaguchi-1.pdf>

第一フラグメント便乗攻撃（1st-fragment Piggybacking Attacks）の概要と対策

第一フラグメント便乗攻撃（1st-fragment Piggybacking Attacks）とは

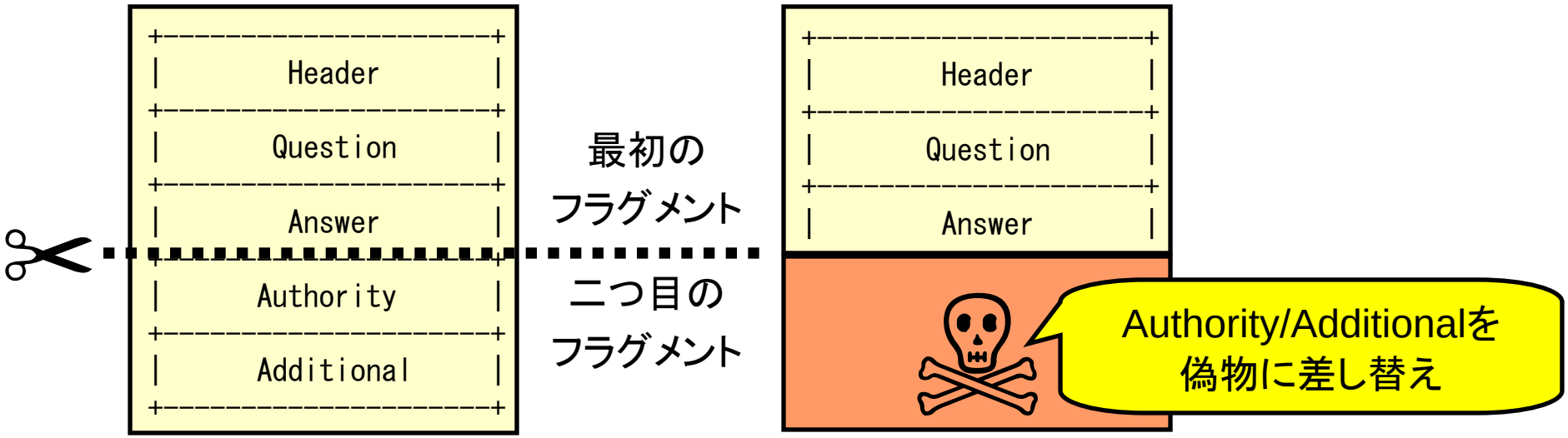
- イスラエル・バル＝イラン大学のAmir Herzberg教授とHaya Shulman氏により発表された論文において初めて報告（2012年5月17日公開）
  - Fragmentation Considered Poisonous
    <http://arxiv.org/abs/1205.4011>
- 発表直後は大きな話題とはならず
  - DNS関係者の間で認識されていなかった可能性あり

第一フラグメント便乗攻撃とは（続き）

- その後、IETF 87 saag（Security Area Advisory Group）の招待講演において、Shulman氏がこの攻撃の詳細を発表（2013年8月1日）
  - DNS Cache-Poisoning: New Vulnerabilities and Implications, or: DNSSEC, the time has come!
    <http://www.ietf.org/proceedings/87/slides/slides-87-saag-3.pdf>
- 発表から1カ月後あたりから、dns-operationsメーリングリスト（dns-oarc.net）で大きな話題に
  - [dns-operations] DNS Attack over UDP fragmentation
    <https://lists.dns-oarc.net/pipermail/dns-operations/2013-September/010625.html>
  - 100通以上の投稿

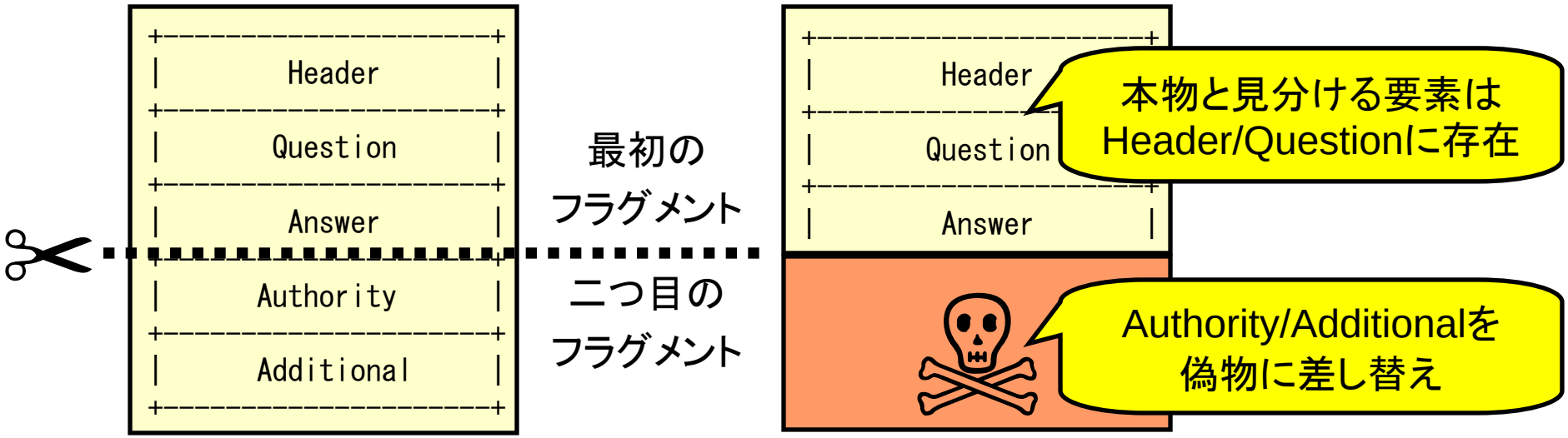
攻撃の概要(1/2)

- IPフラグメンテーションの仕様を悪用した、新たなDNSキャッシュポイズニング攻撃手法
- 攻撃対象:フラグメントされたUDPでのDNS応答
- 応答の二つ目(以降)のフラグメントを偽物に差し替えることで、応答のAuthority/Additionalを偽物に差し替え



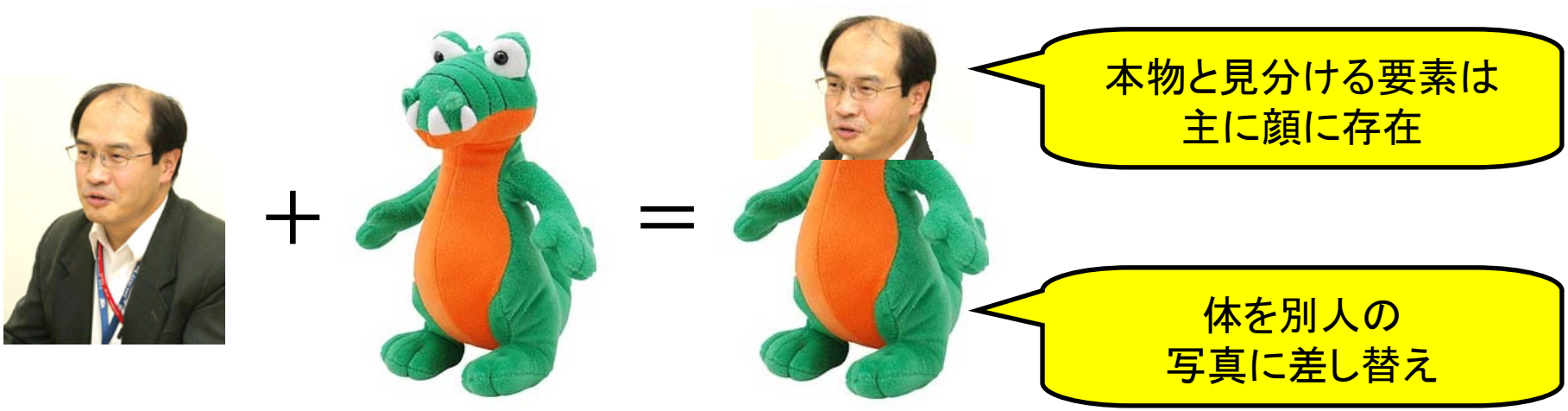
攻撃の概要(2/2)

- DNS応答の同定に使える要素が、最初のフラグメントにしか存在しないことを悪用
  - ポート番号(UDPヘッダー)
  - 問い合わせID、問い合わせ名(Header/Question)
- Authority/Additionalを偽物に差し替えることで、偽の権威DNSサーバー(NS)に誘導させられる危険性あり



コラージュの作成に類似

- コラージュを作成する行為に類似
- 本物と見分ける要素が主に顔にのみ存在することを利用、体を別人の写真に差し替え
  - 顔はむしろ、本物であると判定させることに活用



二つの攻撃手法（アタックベクター）

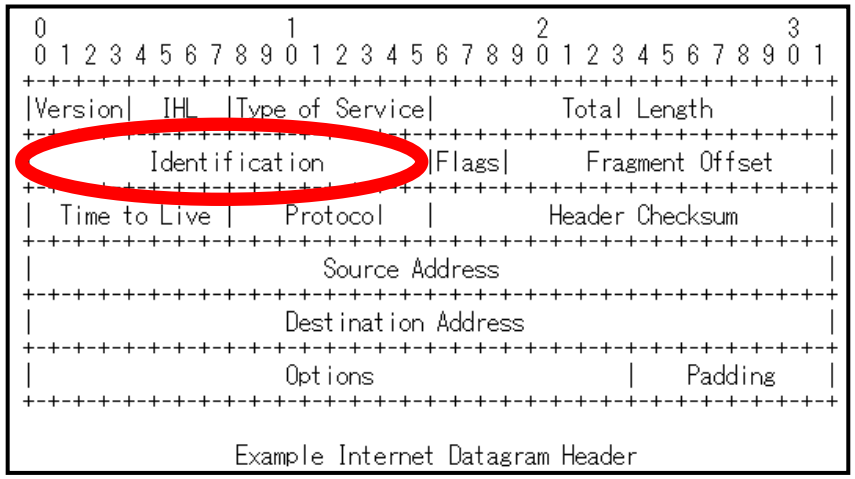
- フラグメントが発生する大きなDNS応答を狙う
  - Shulman氏の論文に書かれている方法
    - DNSSECに対応したドメイン名のDNSKEY RR（必ず使用）
    - 登録済ドメイン名に長い名前のNSを登録
- IPフラグメンテーションを意図的に発生させる
  - 2013年10月のRIPE Meetingにおいて、CZ.NICのT. Hlavacek氏が発表した方法
    - IP fragmentation attack on DNS
      <https://ripe67.ripe.net/presentations/240-ipfragattack.pdf>
  - 偽のICMP PacketTooBigを送り付けてMTUが小さいとOSに誤認させ、応答パケットをフラグメントさせる

IPフラグメンテーションの弱点（1/5）

- DNSパケットの同定に使える要素が、最初のフラグメントにしか存在しない（前述）
  - 問い合わせID
  - 問い合わせポート番号（ポートのランダム化は無力）
  - 問い合わせた名前（0x20の使用は無力）
- これらによるエントロピーの向上は期待できない

IPフラグメンテーションの弱点（2/5）

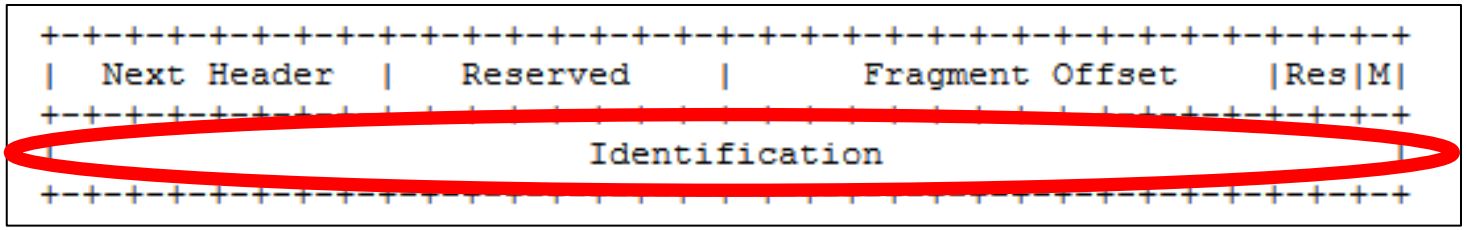
- リアセンブリーにおいて使用されるIdentificationフィールドの大きさが、IPv4では16ビットしかない
  - 二つ目のフラグメントを2の16乗個作成して送り込む、総当たり攻撃が成立しうる
  - ただし、二つ目のフラグメントを大量に送りつけられたOSのプロトコルスタックが具体的にどう振舞うかは未確定



Example Internet Datagram Header（RFC 791より引用）

IPフラグメンテーションの弱点（3/5）

- IPv6ではIdentificationフィールドの大きさは32ビットだが、これだけで安全であるとは言えない（詳細は後述）



IPv6 Fragment Header（RFC 2460より引用）

IPフラグメンテーションの弱点（4/5）

- 二つ目の偽フラグメントを一つ目のフラグメントよりも先に送り込む「先回り」攻撃が可能
  - 攻撃者が攻撃対象の名前解決をトリガーできる場合（オープンリゾルバーや接続先ISPのキャッシュDNSサーバーを攻撃する場合など）において、
  - DNS問い合わせの直後に偽造した二つ目のフラグメントを送り込むことで、
  - 本物のフラグメントよりも（そして、一つ目のフラグメントよりも！）確実に先回りさせられる
  - つまり、攻撃成功（毒入れ）の確率を上げられる

IPフラグメンテーションの弱点（5/5）

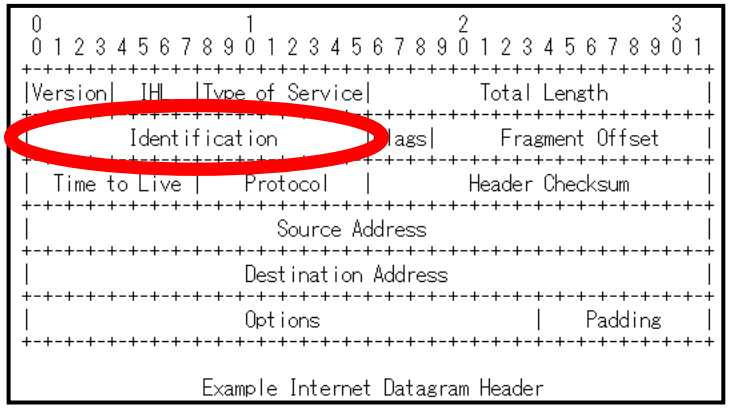
- キャッシュDNSサーバーの応答ログには、攻撃の痕跡が残らない
  - リアセンブリーされなかったフラグメントはIP層で捨てられ、リアセンブリーされた応答のみが上位層に到達する
  - これに対し、カミンスキー型攻撃手法では複数のDNS応答が上位層に到達するため、攻撃の痕跡が残る
- IP/データリンク層には痕跡が残る
  - netstatコマンドやtcpdumpコマンドなどである程度、確認可能

攻撃を成立させるための条件

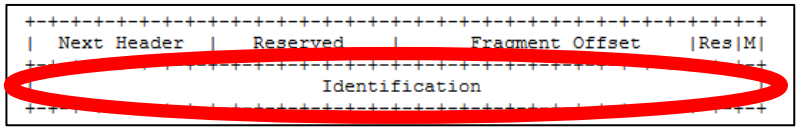
- ip_off（フラグメントオフセット）の確定（予測）
  - 応答の内容とMTUがわかっていれば確定可能
  - コラージュにおける「つなぎ目が不自然にならないようにする」ことに相当
- チェックサムの調整（同じ値になるように）
  - ペイロードやヘッダーの一部を工夫することで調整可能
  - NAT66（RFC 6296）において同様の調整を採用済
  - コラージュにおける「全体として違和感なくつなげられる体（フラグメント）」を準備しておくことに相当
- 上記二つはいずれも、不可能な条件ではない

検討：IPv6なら大丈夫なのか（1/3）

- IPv6ではIPフラグメンテーションにおいて参照されるIdentificationフィールドは前述の通り32ビットであり、総当たり攻撃は一見難しいように思える



RFC 791より引用



RFC 2460より引用

検討：IPv6なら大丈夫なのか（2/3）

- しかし、IPv6の仕様（RFC 2460 4.5 Fragment Header）には以下の記述がある
  - Rather, it is assumed that the requirement can be met by maintaining the Identification value as a simple, 32bit, "wrap-around" counter, incremented each time a packet must be fragmented. It is an implementation choice whether to maintain a single counter for the node or multiple counters, e.g., one for each of the node's possible source addresses, or one for each active (source address, destination address) combination.
- つまり、RFC 2460に従った実装では、Identificationの値は外部から予測可能となる場合がある

検討：IPv6なら大丈夫なのか（3/3）

- 本件については、Fernando Gont氏による調査結果が公開されており、Identificationの値が外部から予測可能な実装が、複数存在していることが明らかになっている
  - “Security Implications of Predictable Fragment Identification Values” Appendix B.
    （draft-ietf-6man-predictable-fragment-id-00）
- OSごとの実装状況（上記I-Dより引用）
  - 予測不能：FreeBSD 9.0、Linux-current、NetBSD 5.1、OpenBSD-current
  - 予測可能：Linux 3.0.0-15、Solaris 10、Windows XP SP2、Windows Vista (Build 6000)、Windows 7 Home Premium
- 予測可能な場合、総当たりが不要になる

検討：DNSSECの導入は有効か（1/2）

- DNSSECの導入により、「コラージュ済み」応答が不正なものであると検知できるようになる
  - キャッシュポイズニング攻撃は成立しなくなる
- 現在のDNSSECの実装では不正な応答を検知した場合、即座にSERVFAILエラーになる
  - 意図的なDoS攻撃（当該の名前へのアクセス妨害）は、現在のDNSSECの実装では防止不可
    - DNSSECの本来の仕様

検討：DNSSECの導入は有効か（2/2）

- かつ、正当な最初のフラグメントは偽のフラグメントと「コラージュされて」しまっているため、正当なDNS応答は到達しなくなっていることにも要注目
- このため、仮に「遅れて到達するであろう正当なDNS応答の到達を待つ」というDNSSECバリデーターの実装が開発されたとしても、正当な名前解決はできなくなる

提案された対策例（IP層における対策）

- IPフラグメントの仕様を拡張する
  - IPv6 Stateless Fragmentation Identification Options（draft-andrews-6man-fragopt）
- コラージュに対し「体（フラグメント）にも顔と同じ効力を持つ目印を付け、受け取り側でチェックする」ことに相当
  - 非現実的

提案された対策例（キャッシュDNSサーバーにおける対策）

- キャッシュポイズニングの影響を小さくする
  - 応答のauthority sectionに設定されたNSレコードのホスト名に、ランダムなprefixを付けてキャッシュする
    - additional sectionに付与されるグルー（A/AAAA）も、それに併せて変更する
  - 上記により、NS/グルーが同一であっても、各委任先ゾーンごとに別扱いになるようにする
- Google Public DNSにおいて採用済
  - カミンスキー型攻撃手法への対策として導入
    - Security Benefits - Public DNS — Google Developers <https://developers.google.com/speed/public-dns/docs/security#nonce_prefixes>

提案された対策例（権威DNSサーバーにおける対策）

- 応答にランダムなRRを付与する
- EDNS0のバッファーサイズを毎回変更する
  - いずれの対策も「コラージュを成立しにくくするため、継ぎ目を一定でなくす」ことに相当

提案された対策例（UDPにおける最大ペイロード長の抑制）

- DNSのUDPにおける最大ペイロード長を、IPフラグメンテーションが発生しない大きさに抑制する
  - max-udp-size（BIND 9）やudp-max-size（Unbound）など
- 現在のデフォルト値はBIND 9/Unboundともに4096
  - DNSSEC（RFC 4035）では「少なくとも1220のサポート必須」「4000をサポートすべき」と規定
- 設定値の参考となるもの
  - 1220：DNSSEC対応リゾルバーにおける、最小のEDNS0バッファーサイズ（RFC 4035）
  - 1280～1410：EDNS0（RFC 6891）における「Ethernetにおけるリーズナブルな値」

提案された対策例（DNSメッセージサイズの抑制）

- DNSSECの鍵や署名をできるだけ短くする
  - 鍵長やロールオーバーの方式を工夫する
    - JP DNSサーバーに設定されるDNSSEC関連情報の内容一部変更について
      <http://jprs.jp/tech/notice/2011-07-29-jpdns-dnskey-and-rrsig-change.html>
  - アルゴリズムとしてECDSA（楕円関数）を使用する
    - すべてのバリデーターが楕円関数に対応しているわけではないことに注意

提案された対策例（Linuxカーネルにおける対策）

- PMTUDの結果を無視するソケットオプションを新設
  - これにより、偽のICMP PacketTooBigを受け入れないように設定可能
- Linux-currentに導入済
  - Linux 3.14においてリリース版にも導入予定
- 想定される使用方法
  - max-udp-size 1220などによりフラグメントが起こらない状態に設定
  - かつ、このソケットオプションを使用し、偽のICMP Packet too Bigを受け入れないように設定
- 参考：dns-operations ML投稿されたサマリー
  - [dns-operations] summary of recent vulnerabilities in DNS security.
    <https://lists.dns-oarc.net/pipermail/dns-operations/2014-January/011249.html>

IPフラグメンテーションの回避による影響(1/2)

- IPフラグメンテーションを回避する設定を各DNSサーバーに適用した場合、TCPの問い合わせ数の増加が予想される
- ルートサーバーやJP DNSサーバーなどの負荷増大につながる
  - どの程度増加するのかについては検証が必要
- IP Anycastの運用に影響を与える可能性がある
  - どの程度影響があるのかについては検証が必要

IPフラグメンテーションの回避による影響(2/2)

- ホームルーターなどのDNSプロキシーにおいて、TCPフォールバックを正しくハンドリングできる必要がある
  - 特に、DANEやDNSSEC検証をクライアント(Webブラウザー)側で実施する場合に必須
- 今回の問題にかかわらず、本来対応が必要
  - IPフラグメンテーションに対応できないホームルーターやDNSプロキシーも数多く存在

まとめ：第一フラグメント便乗攻撃（1/4）

- IPフラグメンテーションの仕様を悪用
  - DNS応答の二つ目（以降）のフラグメントを差し替え
- DNS応答の同定に使える要素を無効化
  - 同定に使える要素は最初のフラグメントに存在
- 二つの攻撃手法（アタックベクター）
  - 大きなDNS応答を狙う
    - DNSKEY RR、長い名前のNSレコード
  - IPフラグメンテーションを意図的に発生させる
    - 偽のICMP PacketTooBigを送付

まとめ：第一フラグメント便乗攻撃（2/4）

- Identificationフィールド（リアセンブリーで使用）
  - IPv4：16ビット
    - 総当たり攻撃に対して脆弱
    - 攻撃の効果はOSのプロトコルスタックの振舞いにも依存
  - IPv6：32ビット
    - ただし、外部から予測可能な実装が存在
    - この場合、総当たり攻撃が不要になる
- 「先回り」攻撃が可能
  - 二つ目のフラグメントを先に送りつけることで、攻撃成功の確率を上げられる

まとめ：第一フラグメント便乗攻撃（3/4）

- 提案された対策
  - DNSSECの導入
  - IPフラグメンテーションの仕様拡張
  - DNSサーバー実装における工夫
    - キャッシュDNSサーバー
    - 権威DNSサーバー
  - IPフラグメンテーションの回避
    - UDPにおける最大ペイロード長の抑制
    - DNSメッセージサイズの抑制
    - OSカーネル（プロトコルスタック）における対策

まとめ：第一フラグメント便乗攻撃（4/4）

- IPフラグメンテーションの回避による影響
  - TCPの問い合わせ数の増加
  - ルート/TLDのサーバーの負荷増大
  - IP Anycastの運用への影響の可能性
- TCPフォールバックへの対応
  - EDNS0への対応とともに今回の問題にかかわらず、本来対応が必要

Q and A / discussion