CDNを回避する攻撃のDRDoSハニーポットによる実態調査

西添 友美[1,a)]　牧田 大佑[2,1]　吉岡 克成[3,4]　松本 勉[3,4]　Michel van Eeten[4,5]

概要：DDoS 攻撃対策として CDN が注目を浴びている．ところが，CDN で防御されているはずの配信元サーバを特定できる場合があることが指摘されている．本研究では，CDN を回避して配信元サーバを直接狙う攻撃の実態を調査した．まず，配信元サーバを特定するツールを分析したところ，いずれもサブドメインの名前解決により配信元サーバを探索していることがわかった．次に，DRDoS ハニーポットが 2016 年 1 月から 7 月までに観測した DRDoS 攻撃を調査した結果，2,089 件の攻撃が配信元サーバ宛であった．CDN を導入するだけでなく，配信元サーバの特定を防ぐことが DDoS 攻撃対策には不可欠である．

キーワード：CDN，DRDoS 攻撃，DRDoS ハニーポット

1. はじめに

近年，Booter や Stresser と呼ばれる DDoS 攻撃代行サービスが手頃な価格で利用できるようになり，DDoS 攻撃の大衆化が進んでいる．日本国内では，オンラインゲームに DDoS 攻撃を実行したとして熊本市の男子高校生が 2014 年 9 月に書類送検された [1]．その高校生は月額 8 ドルで利用が可能な海外の DDoS 代行サービスを用いて DDoS 攻撃を行ったという．このように，DDoS 攻撃を実行するのに必要とされる高い技術を持たなくても，誰でも DDoS 攻撃を実行でき，攻撃対象に被害をもたらすことが可能な状況になってきている．

多くの DDoS 攻撃代行サービスが採用している DDoS 攻撃の手法の 1 つに Distributed Reflection Denial-of-Service（DRDoS）攻撃がある．DRDoS 攻撃はインターネット上に公開されている様々なサービスを経由して通信量を増幅させ，攻撃対象の通信帯域を圧迫することで，サービスの提供を妨害する攻撃である．比較的少量の通信を送信するだけで膨大な量の通信を生成できることから，近年では DRDoS 攻撃による被害が後を絶たず，深刻な脅威となっている．

DRDoS 攻撃を含む DDoS 攻撃への対策が求められる中，Contents Delivery Network（CDN）が注目を浴びている．本来，CDN はコンテンツ配信用ネットワークである．キャッシュサーバを世界中に分散して配置し，エンドユーザから近く，かつ低負荷なサーバからキャッシュされたコンテンツを配信することで，通信の高速化や高い可用性を実現している．

こうした CDN の仕組みは，Web サイトの DDoS 攻撃対策としても有効である．オンプレミス型の DDoS 攻撃防御装置は捌ききれる通信帯域に限界があるため，DRDoS 攻撃のような帯域消費型の DDoS 攻撃は防げない．一方で CDN の場合は，エンドユーザが直接アクセスするキャッシュサーバが世界中に分散して配置されているため，DDoS 攻撃の標的となっても攻撃通信が分散される．すると個々のキャッシュサーバに到達する攻撃通信の量は少なくなり，CDN 側で攻撃通信を充分吸収することが可能になる．さらに，攻撃の被害を受けていないキャッシュサーバからコンテンツの配信を継続することも可能である．

Web サイトの DDoS 攻撃対策として CDN を導入する上で弱点となるのは，オリジンサーバと呼ばれるコンテンツの配信元サーバである．エンドユーザから Web サイトへの通信を CDN 経由に切り替えるために，CDN を導入

[1] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University
[2] 情報通信研究機構
National Institute of Information and Communications Technology
[3] 横浜国立大学大学院環境情報研究院
Graduate School of Environment and Information Sciences, Yokohama National University
[4] 横浜国立大学先端科学高等研究院
Institute of Advanced Sciences, Yokohama National University
[5] デルフト工科大学
Delft University of Technology
[a)] nishizoe-tomomi-ny@ynu.jp

する際には DNS の設定を変更する必要がある．この設定により，Web サイトのドメイン名を名前解決すると CDN のキャッシュサーバの IP アドレスが返ってくるようになる．したがって，通常はオリジンサーバの IP アドレスを知ることができない．ところが，Web サイトの構築方法の不備により，オリジンサーバの IP アドレスが漏洩される場合がある．もしオリジンサーバの IP アドレスが特定され，CDN を回避して直接 DDoS 攻撃を受ければ，CDN のキャッシュサーバでキャッシュ切れが発生し，エンドユーザへのコンテンツ配信ができなくなる．

文献 [2] は，オリジンサーバの IP アドレスが様々な手法によって特定される可能性があることを指摘している．例えば，オリジンサーバの IP アドレスに名前解決されるサブドメインを探索する手法や，DNS データの履歴を遡る手法が紹介されている．さらに，文献 [3] はこの問題の影響を評価するために 17,877 個のドメインを対象とした調査を行なっており，71.5 %のドメインがオリジンサーバの IP アドレスを特定された．

本研究では，CDN の背後に隠されているはずのオリジンサーバを直接狙う攻撃の実態を調査するために，オリジンサーバ特定ツールと DRDoS ハニーポットの観測データの分析を行う．そうした攻撃の実態が明らかになれば，オリジンサーバの IP アドレスが漏洩する問題への対策が促進されるであろう．

一般的に，DoS/DDoS 攻撃は被害者側や通信経路上の ISP でしか観測できないが，DRDoS 攻撃の場合は DRDoS ハニーポットにより第三者からでも観測可能である．DRDoS ハニーポットは，DRDoS 攻撃に悪用されるサービス（DNS，NTP など）を提供するサーバを囮（おとり）としてインターネット上に公開することで，DRDoS 攻撃を観測するシステムである [4], [5]．DRDoS 攻撃はその効率の良さから攻撃者によく用いられる手法であるため，オリジンサーバに対しても DRDoS 攻撃が行われ，その攻撃通信が DRDoS ハニーポットで観測できている可能性がある．

そこで本研究では，我々が運用している DRDoS ハニーポットが観測した DRDoS 攻撃を基に，CDN を回避してオリジンサーバを直接狙う攻撃を調査する．Alexa 上位 100 万サイト [9] から CDN を導入している Web サイトを収集し，それらのサブドメインを名前解決することでオリジンサーバの IP アドレスを探索した．CDN を導入しているドメインは 104,981 個あり，そのうちオリジンサーバが特定されたドメインは 32,290 個（30.8 %）あった．2016 年 1 月から 7 月までの期間に 7 台の DRDoS ハニーポットで観測された 919,731 件の DRDoS 攻撃のうち，今回発見したオリジンサーバを標的とした DRDoS 攻撃は 2,089 件あり，標的となっていた可能性のあるドメインは 2,599 個あった．

2. DRDoS 攻撃の観測

2.1 DRDoS 攻撃

DRDoS 攻撃はインターネット上に公開されている様々なサービスを経由して通信量を増幅させ，攻撃対象の通信帯域を圧迫することで，サービスの提供を妨害する攻撃である．踏み台にされるサーバはリフレクタと呼ばれ，反射効果と増幅効果を持つサービスが攻撃に悪用される．

反射効果とは，要求パケットの送信元 IP アドレスが正しいかを検証せずに，その IP アドレスに対して応答パケットを送信する性質である．DRDoS 攻撃では，攻撃者がリフレクタに対して要求パケットを送信する際に，送信元に攻撃対象の IP アドレスを指定する．攻撃対象に届く応答パケットの送信元はリフレクタの IP アドレスになるため，攻撃対象側から攻撃者を特定することは困難である．

増幅効果とは，要求よりも応答のほうが通信量が大きくなる性質である．この性質により，攻撃者自身が送信する通信量は比較的少量でも，リフレクタから攻撃対象に届く通信量は膨大な量となる．

DRDoS 攻撃に悪用可能なプロトコルは複数種類存在している．文献 [7] では UDP ベースのプロトコルから DNS，NTP，SNMP などの 14 種類が DRDoS 攻撃に悪用可能だと指摘している．また，文献 [8] は TCP の 3WAY ハンドシェイクも悪用できることを示している．

2.2 DRDoS ハニーポット

DRDoS ハニーポットは，DRDoS 攻撃に悪用されるサービス（DNS，NTP など）を提供するサーバを囮（おとり）としてインターネット上に公開することで，DRDoS 攻撃を観測するシステムである [4], [5]．設置からしばらくすると攻撃者からのスキャンが来る．その後，攻撃者が DRDoS ハニーポットをリフレクタとして DRDoS 攻撃に悪用するようになり，DRDoS ハニーポットでは送信元が攻撃対象の IP アドレスに詐称された大量のリクエストパケットが届くようになる．当然，攻撃に加担しないために，DRDoS ハニーポットから外部へ送信する通信量は適切に制限する．

3. CDN を回避する攻撃

DDoS 攻撃対策として Web サイト全体を CDN 経由で配信する場合，CDN で攻撃通信を吸収し，オリジンサーバ（配信元サーバ）に直接攻撃させないことが重要となる．DNS によって Web サイトのドメイン名から CDN のキャッシュサーバへ誘導することで，オリジンサーバの IP アドレスを秘匿している．ところが，この誘導が有効なのはドメイン名を指定した通信のみである．もし攻撃者がオリジンサーバの IP アドレスを知ることができれば，オリジンサーバに対して直接攻撃を行うことが可能となる．

CDN に守られた Web サイトのオリジンサーバの IP アドレスを特定する手法はいくつか報告されている [2], [3]. その中で最も実装が簡単で，かつ特定成功率が高いのはサブドメインの名前解決による手法である [3].

CDN は同時に複数のドメインのリバースプロキシとして動作するために，HTTP リクエストヘッダの `Host` フィールドを基に転送先のドメインを判断している．したがって，オリジンサーバの FTP や SSH に対して CDN 経由で接続することはできない．そこで，オリジンサーバを管理しやすくするために，`ftp.example.com` や `ssh.example.com` など，Web サイトのドメインのサブドメインにオリジンサーバの IP アドレスを割り当てることがある．DNS の仕組み上，あるドメインの全てのサブドメインを知ることは基本的に困難であるが，サブドメインを推測したり，よく使用されるサブドメインの一覧を順番に名前解決することで，オリジンサーバの IP アドレスが割り当てられたサブドメインを発見できる場合がある．

4. オリジンサーバ特定ツールの分析

“CloudFlare Resolver” というキーワードで Web 検索すると，CDN の 1 つである CloudFlare を導入した Web サイトのオリジンサーバの IP アドレスを特定することを謳うツールを多数見つけることができる．また，同じキーワードで YouTube を検索すると，そうしたツールの使用方法や効果を解説した多数の動画が見つかる．さらに，DDoS 攻撃代行サービスの機能の 1 つとして提供されている場合もあり，このようなツールで特定したオリジンサーバの IP アドレスに対して，DDoS 攻撃代行サービスを用いて DDoS 攻撃を実行することが簡単にできる状況となっている．原理的には CloudFlare 以外の CDN にも適用可能であるため，こうしたツールをオリジンサーバ特定ツールと呼ぶことにする．

本章ではオリジンサーバ特定ツールを分析し，それらがどのような方法でオリジンサーバを探索しているかを明らかにする．

4.1 分析方法

オリジンサーバ特定ツールの形態は，Web アプリケーションとして提供されている場合と，ローカル環境上で実行する場合の 2 種類に大別される．本研究では，上記の 2 種類に分け，それぞれ別の方法で分析を行った．

4.1.1 Web アプリケーションとして提供されているツール

Web アプリケーションとして提供されているツールの場合，Web ページ上で探索対象のドメイン名や URL を入力してボタンを押下すると，サーバ側がオリジンサーバを探索するスクリプトを実行し，その結果が Web ページに表示される．本研究では 9 種類のツールを分析対象とし，用意したドメイン名を入力することで各ツールの挙動を外部から観測した．

実験環境は権威 DNS サーバと Web サーバの 2 台で構成される．権威 DNS サーバには NSD[10] を，Web サーバには Nginx[11] を用い，インターネット上の任意のホストからアクセスできるように各サーバを設置した．各サーバの通信ログは tcpdump コマンドにより取得した．

本実験用にドメインを 2 つ取得し，それぞれ Web アクセス観測用と DNS クエリ観測用とした．

まず，CloudFlare[12] の無償プランを契約し，案内に従って Web アクセス観測用ドメインのグルーレコードを CloudFlare の権威 DNS サーバに変更した．CloudFlare の DNS の設定において，Web アクセス観測用ドメインの `www` サブドメインの A レコードを CloudFlare を経由するように設定し，オリジンサーバには用意した Web サーバの IP アドレスを指定した．また，オリジンサーバの IP アドレスの漏洩を再現するために，`ftp` サブドメインの A レコードに Web サーバの IP アドレスを設定した．

次に，DNS クエリ観測用ドメインのグルーレコードに用意した権威 DNS サーバの IP アドレスを指定し，DNS クエリ観測用ドメインの問い合わせを全て観測できるようにした．CloudFlare を導入した Web サイトの DNS レコードを模擬するために，`www` サブドメインの A レコードには CloudFlare のキャッシュサーバの IP アドレスを，`ftp` サブドメインの A レコードには Web サーバの IP アドレスを指定したゾーンファイルを作成した．

各ツールの入力欄に上記 2 種類のドメインを入力し，ボタンを押下してから結果が表示されるまでの間の通信を各サーバで観測した．なお，入力されたドメイン名に対して単純にホスト名を付加してサブドメイン名を生成するツールがあるため，`www` サブドメインと，ホスト名を含まない Apex ドメインの両方を入力した．

4.1.2 ローカル環境で実行するツール

ローカル環境上で実行するツールとして代表的な WebSploit Framework[13] の `web/cloudflare_resolver` モジュールと，Nmap[14] の `dns-brute` スクリプトを分析対象とした．各ツールのドキュメントやソースコードを参照し，どのようにオリジンサーバを探索するか調査した．

4.2 分析結果

オリジンサーバ特定ツールの分析結果を**表 1** に示す．WebSploit と Nmap を除き，悪用防止の観点からドメイン名をもとに各ツールに固有の 3 レターコードを割り当てた．SKR，ORC，IHT の 3 つについては，ボタン押下後に権威 DNS サーバでクエリが観測されず，ページ上にエラーが表示されたか，もしくは何も表示されなかった．また，DGO はボタン押下後に `Nmap` の `dns-brute` モジュールの出力が表示された．Nmap についてはホスト名を列挙したファイルをオプションで指定することが可能であった

表 1　オリジンサーバ特定ツールが名前解決したサブドメインの数.

ツール	実行環境	名前解決するサブドメインの数
IHI	Web	12
SKR	Web	0
SIR	Web	24
DGO	Web	127
ORC	Web	0
EXA	Web	18
ITH	Web	0
GOD	Web	41
VBT	Web	6
WebSploit	ローカル	30
Nmap	ローカル	127

が，デフォルトで指定されているリスト [17] に列挙されているホスト名を数えた．

SKR，ORC，IHT を除いた全てのツールはサブドメインの名前解決を行なっていた．8 個全てのツールが共通して名前解決していたホスト名は，`direct`，`ftp`，`mail` の 3 種類であった．続いて，7 個のツールは `forum` を，6 個のツールは `server`，`dev`，`cpanel`，`blog`，`beta`，`admin` を，5 個のツールは `www`，`ssl`，`help`，`dns`，`direct-connect` を名前解決していた．

SIR と VBT の 2 つについては，結果の画面で IP アドレスの隣に “(Not Cloudflare)” と表示しており，サブドメインの探索によって得られた IP アドレスが CloudFlare ではないこと，すなわちオリジンサーバであることを検証しているように見えるが，CloudFlare のキャッシュサーバの IP アドレスも “(Not Cloudflare)” と表示してしまっていた．

Web アプリケーションとして提供されている 9 つのツールを実行した際に，いずれもオリジンサーバとして用意した Web サーバへのアクセスは観測されなかった．また，ローカル環境で実行する 2 つのツールに関しても，Web サーバへアクセスする機能は無かった．

5. オリジンサーバを狙った DRDoS 攻撃の実態調査

CDN を回避し，オリジンサーバを直接狙った DRDoS 攻撃が実際にどの程度発生しているかを明らかにするため，我々が運用している DRDoS ハニーポットの観測データを用いて調査を行った．

5.1 DRDoS ハニーポットの観測データ

我々は日本国内の ISP に複数台の DRDoS ハニーポットを設置し，継続的に DRDoS 攻撃の観測を行っている．本研究では，2016 年 1 月 1 日から 2016 年 7 月 31 日までの 7 ヶ月間（213 日）に，7 台の DRDoS ハニーポットが観測した 919,731 件の DRDoS 攻撃を調査対象とした．サービス別の攻撃件数は，DNS（53/UDP）が 545,700 件（59.3 %），NTP（123/UDP）が 263,225 件（28.6 %），CharGen（19/UDP）が 81,505 件（8.9 %），SNMP（161/UDP）が 40,363 件（4.4 %），SSDP（1900/UDP）が 7,013 件（0.8 %），QOTD（17/UDP）が 56 件（0.006 %）であった．

5.2 オリジンサーバの IP アドレスの収集

5.2.1 CDN を導入しているドメインの収集

Alexa top sites[9] の上位 100 万サイトから，Web サイト全体を CDN で配信しているドメインを収集した．調査対象の CDN は，Akamai，Amazon CloudFront，CDNetworks，ChinaNetCenter，CloudFlare，Edgecast，Fastly，Incapsula，Level 3，Limelight の 10 社とした．ドメインの収集手順を以下に示す．

(1) 2016 年 7 月 14 日に取得した Alexa top sites の上位 100 万サイトの全てについて，Apex ドメインと `www` サブドメインの A レコードを問い合わせた．名前解決には dig コマンドと，Unbound[15] により構築した自前のキャッシュ DNS サーバを用いた．名前解決は 2016 年 7 月 17 日に行った．

(2) Web サイト全体を CDN で配信するために，多くの CDN では Web サイトのドメインに CNAME レコードを設定し，名前解決要求を CDN の権威 DNS サーバに誘導している．Web サイトがどのような技術やサービスを使用しているのか解析するサービスである WebPagetest[16] は，CNAME レコードのドメインが各 CDN に固有のものであるという特徴を基に CDN を判別している．WebPagetest のドメイン-CDN 対応リスト [18] と手順 1 で収集した CNAME レコードを突き合わせ，各ドメインの CDN を判別した．

(3) Amazon CloudFront，CloudFlare，Fastly の 3 社は CDN のキャッシュサーバの IP レンジリストを公開している [19], [20], [21]．手順 2 で CDN を判別できなかったドメインの A レコードと上記の IP レンジリストを比較した．この判別方法は次の 2 つのケースで有用である．

- CloudFlare や Amazon CloudFront の一部の顧客は，Web サイトのドメインの NS レコードを CDN の権威 DNS サーバに変更し，Web サイトのドメインの A レコードが直接 CDN のキャッシュサーバの IP アドレスを指す設定にしている．
- Apex ドメインに CNAME レコードを割り当てられない制約がある [26] ため，Apex ドメインに Fastly を導入したい場合は，Apex ドメインの A レコードに Fastly の Anycast IP アドレスを設定するように案内されている [27].

5.2.2 オリジンサーバの探索

前節で収集したドメインのサブドメインの A レコード

を問い合わせ，得られた IP アドレスをオリジンサーバの候補とした．名前解決したサブドメインは，ftp，direct，mail，webmail，smtp，cpanel，webdisk，whm の 8 種類である．これらは，文献 [3] の調査で最もオリジンサーバを漏洩していたと報告されているサブドメインである．この名前解決処理も 2016 年 7 月 17 日に行った．

5.2.3 オリジンサーバ候補の検証

次に示す手順で，前節で収集したオリジンサーバの候補が本当にオリジンサーバであるかを検証した．

(1) サブドメインにも CDN の IP アドレスが割り当てられている場合がある．そのような場合を除外するため，オリジンサーバ候補の IP アドレスが CDN のキャッシュサーバでないことを確認した．具体的には，以下を判断材料とした．

CNAME レコード　サブドメインを名前解決した際に取得した CNAME レコードに対して，5.2.1 節で使用したドメイン-CDN 対応リストを突き合わせ，一致する CDN が無いか確認した．

IP レンジ　オリジンサーバ候補の IP アドレスが，5.2.1 節で使用した 3 社の IP レンジリストに含まれていないか確認した．

PTR レコード　オリジンサーバ候補の IP アドレスを逆引きし，得られたドメイン名が CDN のものでないか確認した．我々の経験則から，*.akamaitechnologies.com は Akamai，*.r.cloudfront.net は Amazon CloudFront，*.ip.incapdns.net は Incapsula，*.llnw.net は Limelight のキャッシュサーバであると判定した．

組織名　オリジンサーバ候補の IP アドレスが CDN の所属でないことを確認した．組織名情報として，MaxMind 社の GeoIP データベース [22] を用いた．

(2) サブドメインを持っていた Web サイトについて CDN 経由でドキュメントにアクセスし，ランディングページの URL と HTML ソースを取得した．Web ブラウザには Mozilla Firefox を用い，Selenium[23] で巡回を自動化した．実ブラウザを使用した理由を次に示す．

- HTTP ステータスコードによるリダイレクトだけでなく，meta タグや JavaScript によるリダイレクトにも追従するため．
- 人間が操作するブラウザらしく振る舞うアクセスだけを Web Application Firewall（WAF）が許可する場合があるため．例えば，CloudFlare には DDoS 攻撃防御機能として “I’m Under Attack” モードがある [28]．このモードが有効にされている Web サイトにアクセスすると，中間ページが 5 秒間表示された後に要求したページが表示される．この機能は，JavaScript と Cookie を用いてブラウザを検証している．

なお，CloudFlare の WAF により reCAPTCHA[24] を解くことが求められる Web サイトに関しては，後でまとめて手動で巡回した．

(3) 手順 2 で得たランディングページの URL のスキーム，ホスト名，パスを用いてオリジンサーバの候補の IP アドレスに HTTP リクエストを送信し，HTML ソースを取得した．例えば手順 2 で得た URL が https://blog.example.com/about.html，候補 IP アドレスが 10.0.0.1 であれば，URL に https://10.0.0.1/about.html を，HTTP リクエストヘッダの Host フィールドに blog.example.com を指定して HTTP GET リクエストを送信した．HTTP 通信の送受信には cURL コマンドを用いた．

(4) もし候補の IP アドレスが真にオリジンサーバであれば，手順 2 と 3 で取得した Web ページは一致するはずである．手順 2 と 3 で得た HTML ソースから title タグの中身を抽出して比較し，一致した場合は，その IP アドレスはオリジンサーバであるとした．

5.3 調査結果

調査結果を**表 2** にまとめる．

Alexa top sites の上位 100 万サイトのうち，調査対象とした 10 社の CDN を導入していたドメインは 104,981 個であり，その 75.3 %は CloudFlare が占めていた．175 個のドメインは Apex ドメインと www サブドメインがそれぞれ別の CDN を導入していた．

1 種類以上のサブドメインを持っていたドメインは 104,139 個あった．それらから 5.2.3 節の手順 1 で CDN を指すサブドメインを除外すると，67,005 個のドメインが残った．

5.2.3 節によりオリジンサーバの IP アドレスが特定されたドメインは 32,290 個（30.8 %）あった．オリジンサーバの特定に最も貢献したのは ftp サブドメインで，23,094 個のドメインのオリジンサーバの IP アドレスを晒していた．以降は mail（18,469），webdisk（15,267），webmail（14,027），cpanel（13,556），whm（13,295），direct（9,473），smtp（7,997）の順であった．なお，Apex ドメインがオリジンサーバの IP アドレスを指しているケースは 12,155 件あった．

オリジンサーバの IP アドレスは合計で 25,352 個であった．オリジンサーバが特定されたドメインの数よりも少ないのは，Web ホスティングサービスの影響である．オリジンサーバの IP アドレス 1 個あたり平均で 1.4 個のドメインが対応付けられており，2 個以上のドメインが紐付けられた IP アドレスは 3,202 個あった．最も多くのドメインに対応付けられていたのは GitHub Pages の IP アドレスで，460 個のドメインが割り当てられていた．

7 ヶ月間の DRDoS ハニーポットの観測データのうち，オリジンサーバを直接狙った攻撃は 2,089 件発生していた．

オリジンサーバが攻撃を受けていた可能性のあるドメインは 2,599 個あり，攻撃を受けたオリジンサーバの IP アドレスは 825 個あった．

1 ドメインあたりのオリジンサーバが直接攻撃を受けた件数は平均 4.7 件であり，最大値は 195 件であった．また，2 回以上オリジンサーバに攻撃を受けたドメインは 1,099 個あった．

オリジンサーバの IP アドレス 1 個あたりの被攻撃件数は平均で 2.5 件であり，OVH SAS の IP アドレスに対する攻撃件数が 195 件で最大値であった．攻撃を受けたオリジンサーバの IP アドレスが所属する AS（Autonomous System）単位で見ると，OVH SAS が最も多く攻撃を受け（374 件），次いで GoDaddy.com LLC であった（153 件）．オリジンサーバが OVH SAS にあり，かつ攻撃を受けていたドメインの数は 60 個であり，GoDaddy.com LLC の場合は 256 個であった．

オリジンサーバを直接狙った DRDoS 攻撃について悪用されたプロトコル別に集計すると，1385 件（66.3 %）は DNS（53/UDP），454 件（21.7 %）は NTP（123/UDP），246 件（11.8 %）は CharGen（19/UDP），63 件（3.0 %）は SNMP（161/UDP），24 件（1.2 %）は SSDP（1900/UDP）を悪用していた．2 種類以上のプロトコルを悪用した攻撃は 78 件（3.7 %）であった．

オリジンサーバに対する DRDoS 攻撃の継続時間は，1 分以下が 477 件（22.8 %），5 分以下が 1,165 件（55.8 %），10 分以下が 1,415 件（67.7 %），15 分以下が 1,529 件（73.2 %）であった．

5.4 攻撃事例

CDN を回避してオリジンサーバを直接狙っていた攻撃事例を 1 つ取り上げ，詳細を説明する．

fr■.ovh は Web サイト全体を CloudFlare で配信している．Apex ドメインである fr■.ovh を名前解決すると CloudFlare の IP アドレスが 2 つ（104.24.96.■ と 104.24.97.■）が返ってくる．一方，www.fr■.ovh，ftp.fr■.ovh，mail.fr■.ovh，smtp.fr■.ovh といったサブドメインを名前解決すると 149.202.■.■ が返ってくる．この IP アドレスの割り当て先は，Web ホスティングサービスである OVH 社である．

http://fr■.ovh/にアクセスした際に表示されるページと，HTTP リクエストヘッダの `Host` フィールドを fr■.ovh に書き換えて http://149.202.■.■/にアクセスした際に表示されるページを比較すると，広告の有無に違いが見受けられるものの，ほぼ同じ内容であった．したがって，149.202.■.■ は fr■.ovh のオリジンサーバであるといえる．

7 ヶ月間の DRDoS ハニーポットの観測データのうち，149.202.■.■ に対する DRDoS 攻撃は 195 件あった．1 日に 1〜4 件のペースで 2 日に 1 日は攻撃を受けていた．149.202.■.■ は Web ホスティングサービスのアドレスであり，DNSDB[25] によれば fr■.ovh 以外にも多数のドメインをホスティングしているため，これら全てが真に fr■.ovh を狙った攻撃であるかはわからない．

ところが，明らかに fr■.ovh を狙ったと推測できる攻撃があった．2016 年 2 月 5 日の午前 11 時前に DRDoS ハニーポットが観測した，fr■.ovh を狙ったと思われる攻撃の概要を**表 3** に示す．fr■.ovh のオリジンサーバに対して DNS と CharGen を悪用した DRDoS 攻撃を実行した後，fr■.ovh の名前解決結果（CloudFlare のキャッシュサーバ）に対して DNS の DRDoS 攻撃を行い，再びオリジンサーバを攻撃しているように見受けられる．この一連の攻撃が fr■.ovh を狙っていると考えられる根拠を以下に示す．

- fr■.ovh のオリジンサーバとキャッシュサーバに対し，連続して攻撃が発生していたこと．
- すべての攻撃について，DNS クエリのドメイン名とタイプが同一であったこと．DNS アンプ攻撃に利用されるドメイン名にはバリエーションがある [6] ため，同じドメイン名を問い合わせる攻撃は同じ攻撃ツールからの攻撃であることが予想される．
- 受信した要求パケットの IP ヘッダの TTL 値を見ると，ハニーポット毎に同一であったこと．IP-TTL 値が同じということは，同一ネットワークからパケットが送信された可能性が高いことを意味しており，同一の攻撃者が実行した攻撃であることが推測される．

6. 考察

6.1 オリジンサーバ特定ツール

4 章の結果から，既存のオリジンサーバ特定ツールは複数のサブドメインを名前解決することによってオリジンサーバを探索していることがわかった．オリジンサーバを特定する手法はいくつか報告されているが，それらの中で最も実装が簡単で，しかも最も特定成功率が高い手法はサブドメインの探索である [3]．この理由から，オリジンサーバ特定ツールはサブドメインの名前解決による手法を採用したと考えられる．

また，今回調査したオリジンサーバ特定ツールは，いずれも探索によって得られた IP アドレスが本当にオリジンサーバであるかを検証していなかった．その論拠は以下の 2 つである．

- CloudFlare を導入したドメインを入力して実行した結果，どのツールも探索結果の画面に CloudFlare の IP アドレスも表示してしまっていた．IP アドレスの隣に “(Not Cloudflare)” と表示するツールもあったが，CloudFlare の IP アドレスにもその表示をしていたため，実際には検証していないと推測される．
- オリジンサーバとして用意した Web サーバに対して，ツール実行時にアクセスが来なかった．

表 2　Alexa 上位 100 万サイトのうち，CDN を導入したドメインの数とオリジンサーバの IP アドレスが特定されたドメインの数，オリジンサーバが直接攻撃を受けていた可能性のあるドメインの数，攻撃を受けたオリジンサーバの IP アドレスの数，DRDoS ハニーポットで観測されたオリジンサーバへの攻撃件数．

CDN 名	ドメイン数	オリジンが特定されたドメイン数	オリジンの IP アドレス数	オリジンが攻撃を受けたドメイン数	攻撃を受けたオリジンの IP アドレス数	オリジンへの攻撃件数
Akamai	12,614	3,337	2,061	611	66	103
Amazon CloudFront	4,450	365	371	8	12	67
CDNetworks	373	109	99	3	2	2
ChinaNetCenter	752	305	314	15	17	39
CloudFlare	79,008	26,664	21,847	1,369	708	1,860
Edgecast	871	324	201	42	10	38
Fastly	2,533	631	175	471	16	82
Incapsula	3,905	256	200	9	8	14
Level 3	371	184	121	14	6	7
Limelight	279	122	38	60	5	9
合計	104,981	32,290	25,352	2,599	825	2,089

表 3　fr■.ovh を狙ったと推測される DRDoS 攻撃の概要．149.202.■.■ は fr■.ovh のオリジンサーバの IP アドレスであり，104.24.96.■ と 104.24.97.■ は fr■.ovh を配信している CDN のキャッシュサーバの IP アドレスである．

ハニーポット ID	サービス	攻撃対象	開始時刻	終了時刻	継続時間（秒）	受信パケット数	受信パケットの IP-TTL 値	DNS クエリ
H01	DNS	149.202.■.■	10:49:25	10:50:20	55	536	240	httrack.com ANY
H03	DNS	149.202.■.■	10:49:25	10:50:20	55	486	238	httrack.com ANY
H04	CharGen	149.202.■.■	10:49:51	10:49:56	5	1,537	234	—
H05	CharGen	149.202.■.■	10:49:51	10:49:56	5	373	240	—
H01	DNS	104.24.97.■	10:51:57	10:52:11	14	324	240	httrack.com ANY
H03	DNS	104.24.97.■	10:51:57	10:52:11	14	322	238	httrack.com ANY
H01	DNS	104.24.96.■	10:52:26	10:52:44	18	417	240	httrack.com ANY
H03	DNS	104.24.96.■	10:52:26	10:52:44	18	417	238	httrack.com ANY
H01	DNS	149.202.■.■	10:53:51	10:53:58	7	169	240	httrack.com ANY
H03	DNS	149.202.■.■	10:53:51	10:53:58	7	169	238	httrack.com ANY

5 章の調査結果から，サブドメインの名前解決によって得られた IP アドレスがオリジンサーバではない場合が多数存在することがわかった．そのような IP アドレスを関連アドレスと呼ぶことにすると，能力の高くない攻撃者がオリジンサーバ特定ツールの結果を用いて DDoS 攻撃を試みる場合，真のオリジンサーバと関連アドレスの見分けがつかず，それらを順番に攻撃していくであろう．そういった事例の調査は今後の課題としたい．

6.2　オリジンサーバを狙った DRDoS 攻撃

7 ヶ月間の DRDoS ハニーポットの観測結果のうち 2,089 件がオリジンサーバを狙った DRDoS 攻撃であった（表 2）．したがって，CDN を回避し，オリジンサーバを直接狙う DRDoS 攻撃は実際に発生していることが明らかになった．

ただし，今回は発見が漏れたオリジンサーバが存在する可能性があるため，実際にはより多くの攻撃を DRDoS ハニーポットで観測していたかもしれない．具体的には，以下の箇所で漏れが発生した可能性がある．

- Alexa top sites の上位 100 万サイトから CDN を導入したドメインを収集したとき．CDN の判別方法に漏れがあった可能性がある．
- オリジンサーバの IP アドレスを探索したとき．本研究ではサブドメインによる特定手法のみを採用し，さらに名前解決したサブドメインも 8 種類に絞った．他の特定手法も実施していれば，さらに特定成功率が高まったと予想される．
- 得られた IP アドレスがオリジンサーバであるか検証したとき．CDN 経由で取得した Web ページと，オリジンサーバの候補から直接取得した Web ページのコンテンツが一致するか確かめるために，本研究では `title` タグの中身を比較した．その際に，`title` タグが無い Web ページは調査対象外とした．Web ページの比較方法を工夫すれば，特定成功率が高まったと予想される．

また，多数のドメインが共用している Web ホスティングサービスがオリジンサーバであるケースもあり，上記の攻撃の中には調査対象外のドメインに対する攻撃も含まれている可能性がある．しかし表 3 に示したように，明らかに CDN を導入した Web サイトのオリジンサーバを狙ったと推測される攻撃事例も含まれており，CDN を回避す

る攻撃は実際に発生しているといえる．

6.3 対策

文献 [3], [29] で議論されているように，オリジンサーバ特定の脅威への対策としては，オリジンサーバにおいてCDN 以外からのアクセスを遮断すること，CDN 導入時にオリジンサーバの IP アドレスを変えること，推測可能なサブドメインにオリジンサーバの IP アドレスを割り当てないことが挙げられる．オリジンサーバを特定する手法がいくつか存在する中で，4 章と 5 章の結果を鑑みると，サブドメインによる特定手法への対策が最も急務であるといえる．Web サイト管理者がサブドメインにオリジンサーバの IP アドレスを割り当てる目的は，FTP やメールなどの Web 以外のサービスに接続するためである．したがって，以下のような具体策が考えられる．

- オリジンサーバに接続するための別のドメインを取得する
- オリジンサーバに接続するためのサブドメインを予想されにくいものにする
- オリジンサーバに接続するためにドメイン名を使用せずに，IP アドレスから直接接続する
- オリジンサーバには他のサービスを同居させない

7. まとめと今後の課題

本稿では，DRDoS ハニーポットが 2016 年 1 月から 7 月までに観測した DRDoS 攻撃のデータを基に，CDN を回避してオリジンサーバを直接狙う攻撃の調査を行った．オリジンサーバに対して直接攻撃を実行する事例は実際に発生しており，DDoS 攻撃対策として CDN を導入する際にはオリジンサーバを特定されないよう対策することが必須である．

今後は，サブドメイン以外の手法も実施し，オリジンサーバの発見数を増やしてより多くの攻撃事例を分析したい．また，オリジンサーバが特定可能なドメインを継続して監視し，対策状況を調査する予定である．

謝辞 本研究の一部は文部科学省国立大学改革強化推進事業の支援を受けて行われた．

参考文献

[1] 日本産経新聞：オンラインゲームに DDoS 攻撃「面白いので何度もやった」熊本の高 1 男子を書類送検，入手先 ⟨http://www.sankei.com/affairs/news/140918/afr1409180019-n1.html⟩（参照 2016-08-05）．

[2] Nixon, A. and Camejo, C.: DDoS Protection Bypass Techniques, *Black Hat USA* (2013).

[3] Vissers, T., Van Goethem, T., Joosen, W. and Nikiforakis, N.: Maneuvering Around Clouds: Bypassing Cloud-based Security Providers, *ACM CCS*, pp. 1530–1541 (2015).

[4] Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T., Yoshioka, K. and Rossow, C.: AmpPot: Monitoring and Defending Amplification DDoS Attacks, *RAID* (2015).

[5] 牧田大佑，西添友美，小出 駿，筒見拓也，金井文宏，森 博志，吉岡克成，松本 勉，井上大介，中尾康二：早期対応を目的とした統合型 DRDoS 攻撃観測システムの構築，2015 年暗号と情報セキュリティシンポジウム予稿集 CD-ROM (SCIS2015)，No. 2A3-1 (2015).

[6] 牧田大佑，吉岡克成，松本 勉，中里純二，島村隼平，井上大介：DNS アンプ攻撃の事前対策へ向けた DNS ハニーポットとダークネットの相関分析，情報処理学会論文誌，Vol. 56, No. 3, pp. 921–931 (2015).

[7] Rossow, C.: Amplification Hell: Revisiting Network Protocols for DDoS Abuse, *NDSS* (2014).

[8] Kührer, M., Hupperich, T., Rossow, C. and Holz, T.: Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks, *WOOT* (2014).

[9] Alexa, available from ⟨http://www.alexa.com/topsites⟩.

[10] Name Server Daemon (NSD), available from ⟨https://www.nlnetlabs.nl/projects/nsd/⟩.

[11] nginx, available from ⟨https://nginx.org/en/⟩.

[12] CloudFlare, available from ⟨https://www.cloudflare.com/⟩.

[13] websploit, available from ⟨https://sourceforge.net/projects/websploit/⟩.

[14] Nmap, available from ⟨https://nmap.org/⟩.

[15] Unbound, available from ⟨https://www.unbound.net/⟩.

[16] WebPagetest, available from ⟨https://www.webpagetest.org/⟩.

[17] nmap/vhosts-default.lst, GitHub, available from ⟨https://github.com/nmap/nmap/blob/master/nselib/data/vhosts-default.lst⟩

[18] webpagetest/cdn.h, GitHub, available from ⟨https://github.com/WPO-Foundation/webpagetest/blob/fca222034e8a16db17979f824b60009b39fd6650/agent/wpthook/cdn.h⟩.

[19] Locations and IP Address Ranges of CloudFront Edge Servers, Amazon CloudFront, available from ⟨http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/LocationsOfEdgeServers.html⟩.

[20] CloudFlare IP Ranges, CloudFlare, available from ⟨https://www.cloudflare.com/ips/⟩.

[21] Accessing Fastly's IP ranges, Fastly Help Guides, available from ⟨https://docs.fastly.com/guides/securing-communications/accessing-fastlys-ip-ranges⟩.

[22] GeoIP2 データベース，MaxMind，入手先 ⟨https://www.maxmind.com/ja/geoip2-databases⟩.

[23] Selenium, available from ⟨http://www.seleniumhq.org/⟩.

[24] reCAPTCHA, available from ⟨https://www.google.com/recaptcha/intro/index.html⟩.

[25] DNSDB, available from ⟨https://www.dnsdb.info/⟩.

[26] RFC1034, available from ⟨https://www.ietf.org/rfc/rfc1034.txt⟩.

[27] Using Fastly with apex domains, Fastly Help Guides, available from ⟨https://docs.fastly.com/guides/basic-configuration/using-fastly-with-apex-domains⟩ (accessed 2016-08-09).

[28] Prince, M.: Introducing: I'm Under Attack Mode, CloudFlare blog, available from ⟨https://blog.cloudflare.com/introducing-im-under-attack-mode/⟩ (accessed 2016-08-09).

[29] Zhang, E.: Cloudflare IP Leakage, Eric Zhang [Xeroday], available from ⟨https://www.ericzhang.me/resolve-cloudflare-ip-leakage/⟩ (accessed 2016-08-09).