Webリソースの完全性保証フレームワーク

慶應義塾大学 中島 博敬(nunnun)

WIDE研究会2015年5月 学生研究発表

Keio University
1858
CALAMVS
GLADIO
FORTIOR

背景

Keio University
1858
CALAMVS
GLADIO
FORTIOR

Web高速化

- Webの高速化手法は大きく2つ
  - 高RTT環境下でパフォーマンスに大きく影響するHTTPのプロトコル的改良
  - クライアントとリソースのネットワーク的距離を短くする

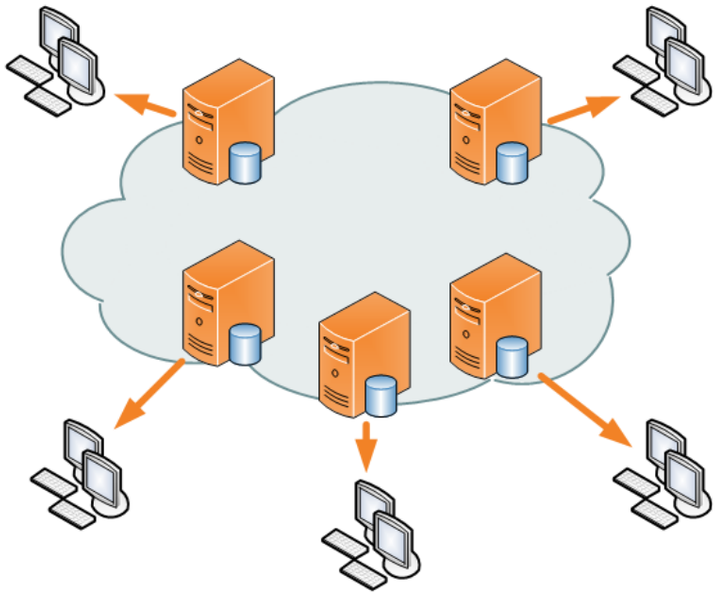
HTTPのプロトコル的改良

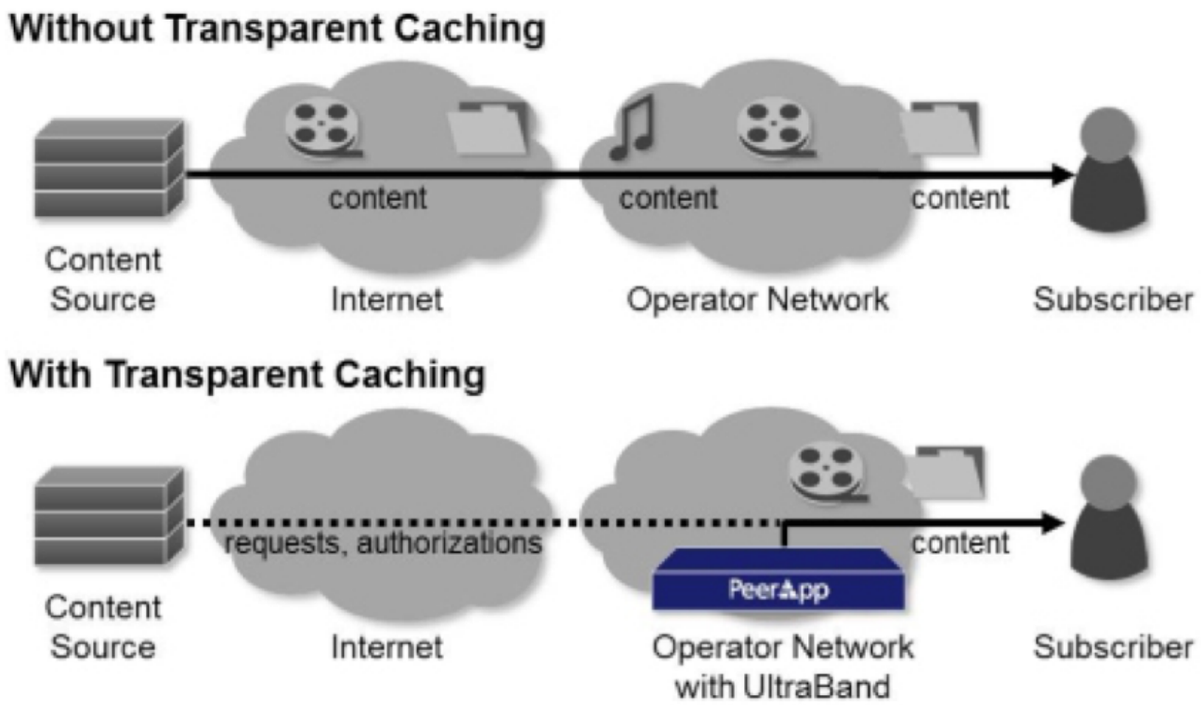
- HTTPの同時接続数制限がモバイルネットワークなど高RTT環境下におけるパフォーマンス低下を招く
- HTTPの同時接続数制限を回避する方法として、プロトコルの改良・Workaroundが提案されている
  - HTTP/1.1 Pipelining
  - SPDY
  - HTTP/2

ネットワーク的距離の短縮

- HTTPサーバ・クライアント間のRTTが高いことが、半二重であるHTTPのパフォーマンス低下の原因
- サーバ・クライアント間のRTTを減らせば、HTTPであっても、パフォーマンス低下は防げる
- 代表的な手法
  - CDN(Contents Delivery Network)
  - Transparent Cache

CDN・Transparent Cache





- CDN
  Webサイトがリソースをネットワークのエッジに配置することで
  ネットワーク的距離を短縮
- Transparent Cache
  ネットワーク側がトラフィックを分析し、リソースが手元のキャッシュ
  サーバにある場合、通信をバイパスすることで擬似的にRTTを短縮

CDN figures from wikipedia: https://en.wikipedia.org/wiki/Content_delivery_network#/media/File:NCDN_-_CDN.png
Transparent Cache figure from CDT: https://cdt.org/blog/open-caching-open-standards-and-privacy/

Transparent Cache事例

- モバイルネットワーク

  各携帯電話ネットワークで導入されている

  - ドコモ(SPモード)
  - KDDI
  - ソフトバンク
- 固定網
  - 国外では事例有り

Transparent Cache事例2

ドコモ

(3) 通信の最適化

① 別途当社の定めるところに従い同意いただいた場合、spモードのアクセスポイントを経由したパケット通信において、画面の表示速度や動画の再生開始時間を早くするための通信の最適化を行う場合があります。最適化とは、端末の画面に適したサイズに画像・動画を圧縮することや、より伝送効率の高いコーデック形式に動画を変換することをいいます。

② HTTPS（Hypertext Transfer Protocol Secure）通信時の画像等、spモード電子メールを含むメールの添付ファイルの最適化は行いません。

③ 最適化された画像等を復元することはできません。

KDDI

通信の最適化について

LTE NETのパケット通信において、以下のファイルを対象に、画面の表示速度や動画の再生開始時間を早くするための通信の最適化を行う場合があります。

画像ファイル	BMP、jpg、gif、PNG形式
動画ファイル	MPEG、AVI、MOV、FLV、MP4、3GP、WebM、ASF、WMV形式

※HTTPS通信、Eメール添付ファイルの最適化は行いません。
※最適化とは、スマートフォンの画面に適したサイズに画像を圧縮・変換することをいいます。なお、圧縮・変換されたデータを復元することはできません。
※通信の最適化を必要とされないお客さまは、お客さまセンター (157) にて非適用のお手続きができます。

ご利用の際に制御することがあるコンテンツ・サービス

対象	説明
VoIP（Voice over Internet Protocol）を利用する通信	音声通話やテレビ電話などをパケット信号に変換し、データ通信にて実現するサービス
動画、画像などの一部	MPEG、AVI、MOV形式などの動画ファイル BMP、JPEG、GIF形式などの画像ファイル
大量のデータ通信、または長時間接続をともなうパケット通信	動画閲覧、高画質画像閲覧をともなうサイト、アプリケーションなど

- 上記コンテンツ・サービスなどをご利用の際、通信速度の制御や各種ファイルの最適化を行う場合があります（最適化されたデータの復元はできません）。なお、通信の切断は行いません。

その他

- その他パケット通信時に、最適化したファイルを機種側で復元できるなど、お客さまの通信に影響がない場合には、各種ファイルの最適化を行います。
- パケット通信時に、コンテンツに表示されない部分のデータなどにつきましては、通常の視聴・閲覧に影響のない範囲でダウンロードの際に省略する場合があります。なお、省略されたデータは復元できません。

SoftBank

JavaScript Library CDN

- jQueryなどJS Libraryの肥大化

  jquery-2.1.4.min.js -> 84KB

- プラグインなどで複数ファイルを読み込む必要性
- CDNを用いるケースが多い
  - cdnjs.com
  - ajax.aspnetcdn.com
  - google code

Webリソース転送における完全性保証

- IPsecにおけるAHにような機能はhttpには存在しない
- TLSではRFC4785(Null Encryption)が存在するが利用されていない
- 一般的な実行コード署名をJavaScriptで行う手法は存在しない
- そのため完全性のみの担保したい場合であってもTLSを使用する場合が散見される
  - JavaScript CDNで使用されているJS等

TLSで保護するのでは不十分な事例

- サーバとクライアント間におけるリソースの完全性が保証
- サーバサイドで改ざんされた場合、確認が困難である
  - WordPressなどで使われるプラグイン配布元で配布していたjQueryが改変された事例
  - メジャーでない脆弱性の場合、検知が難しい
- TLSだけでは配布元が配布したコードなのか確認出来ない
- **TLSの有無に関わらずコードの完全性を確認する方法が必要である**

事前調査

現状の調査

- 配布元と異なるJSライブラリが配布されている実態を調査
- Webサイトで読み込まれるJSと
  公式サイトなどで配布されているJSが異なるかを調査する
- 2種類の方法で調査を行う
  - クローリングによる方法
  - ブラウザ拡張による方法

調査手法

- Webページで読み込まれるJavaScriptファイルについて以下の項目を記録
  - ファイル名
  - ファイルハッシュ値
- これらが公式サイトで配布されている値と異なるか検証
- 対象はJavaScript CDNで配布されているものを対象とする
- 異なる場合はURLと配布されていたJSファイルを保存する

クローリングによる手法

- Alaxa Top 10,000をクローリング
- 既存のクローラではAjaxリクエストなどを処理出来ないため拡張を行う

ブラウザ拡張による手法

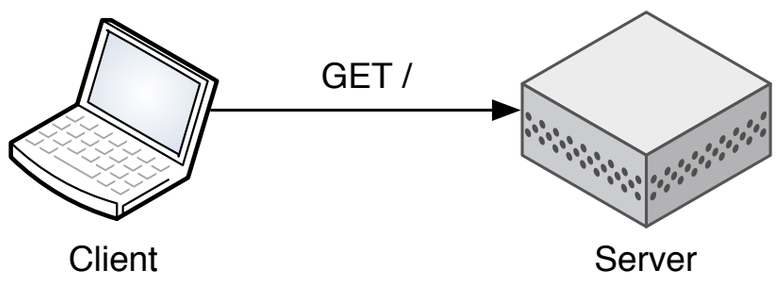
- Webページのリクエストを監視し、
  JavaScriptライブラリであった場合にチェックを行う
- プライバシーの懸念があるため、ファイル名とハッシュ値の適合をクライアント側で行う
- 問題があった場合のみ転送する
- その他にも匿名化処理を実施する

提案手法

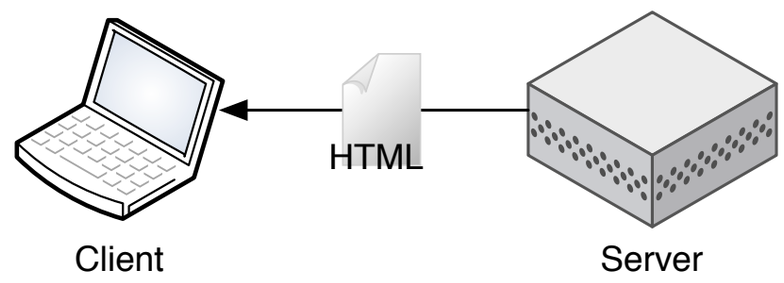
Webリソースの完全性保証フレームワーク

- TLSの有無に関わらずWebリソースの完全性を保証
- 改変を想定しないコードの場合開発元が署名
- Webサーバ側とクライアント側で構成

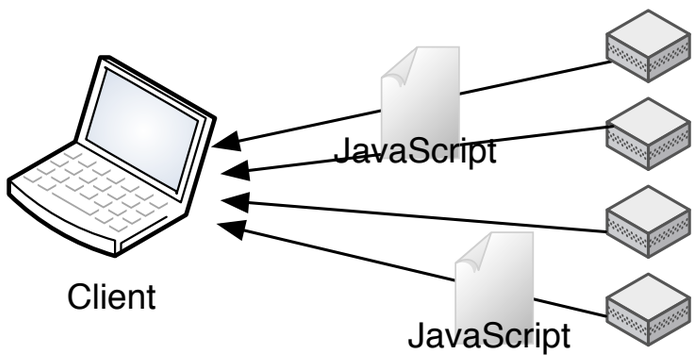
システム概要



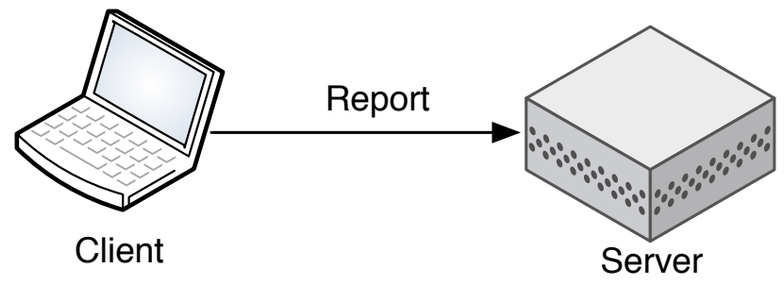
1. クライアントはサーバに対して
ページリクエストを送付する



2. リソースを読み込み署名を検証するコードと署名を
HTMLに埋め込み、返答する

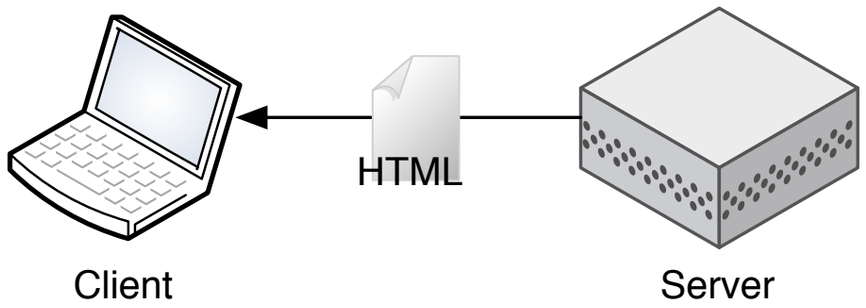


3. JavaScriptコードは指定されたリソースを
サーバもしくは外部サーバより読み込む



4. 署名と異なる場合、
クライアントからサーバ側に通知を行う

2. Bootstrap



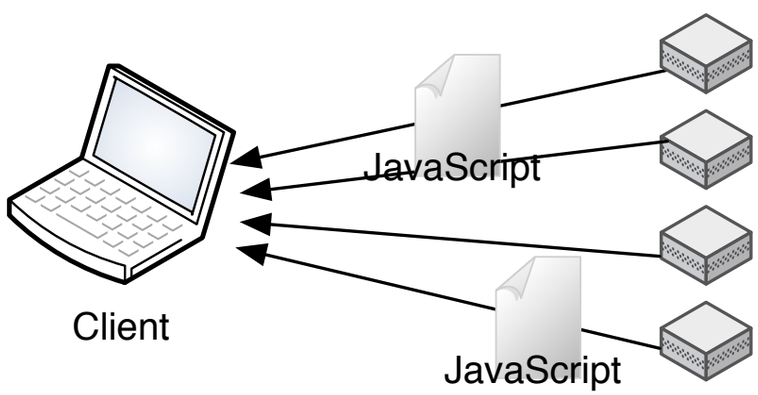
2. リソースを読み込み署名を検証するコードと署名を
HTMLに埋め込み、返答する

- Webサーバは以下のリソースを転送
  - 読み込むJavaScriptのURL
  - そのファイルのハッシュ値・信頼する発行者情報
- これらを読み込み、JavaScriptを読み込むBootstrap
  な役割を果たすJavaScriptも同時に配信される

信頼される発行者とは

- JavaScript CDNなどから最新版のリソースを読み込む場合、ハッシュ値が異なる可能性がある
- そのファイルを配布している人の公開鍵を利用することで、リソースの完全性を確認する

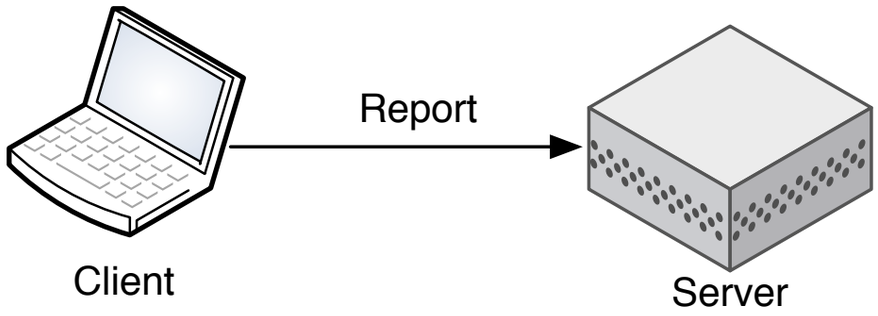
3. JavaScriptの読み込み



3. JavaScriptコードは指定されたリソースを
サーバもしくは外部サーバより読み込む

- Bootstrapは記載されたJavaScriptを読み込み、ハッシュ値・署名の確認を行う
- ハッシュ値・署名と異なる場合、事前に設定された動作を行う（警告の表示、動作の停止など）

4. 改ざん検出時



4. 署名と異なる場合、
クライアントからサーバ側に通知を行う

- 改ざん（改変）を検出した場合
  Webサイトに対して検出を通知する
- 検出を通知することでWebサイト側がいち早く改ざんに対して対策を行うことが可能となる

評価

- サーバ側
  - 署名・ハッシュ値の負荷

    (abによりどの程度パフォーマンスが低下するか検証する)
- クライアント側
  - ページロード時間の変化

    低下することは避けられないがどの程度変化するか評価を行う
- 事前調査で検出した脆弱な事案対してどの程度対処できるか評価を行う

まとめ

- TLSに依存せずWebリソース転送における完全性を確保するためのフレームワークの提案
- JavaScriptを対象にリソースのハッシュ値・署名を用いて検証を行う
- 事前にクローリング・ブラウザ拡張を用いて脆弱な事例を調査し、提案手法でどの程度検出できるか評価を実施する

先行研究

- Measurement
  - You Are What You Include: Large-scale Evaluation of Remote Javascript Inclusions
- HTTP Object Signing
  - Stickler: Defending Against Malicious CDNs in an Unmodified Browser
  - "HTTP signing." U.S. Patent No. 8,677,134