

KOF2014 version



KANSAI OPEN FORUM 2014

関西オープンソース2014 関西コミュニティ大決戦2014

JPCERT  CC[®]

～誰かの失敗を他山の石に～ 脆弱性事例に学ぶセキュアコーディング 「SSL/TLS証明書検証」編

一般社団法人

JPCERTコーディネーションセンター

情報流通対策グループ 脆弱性解析チーム

リードアナリスト 戸田 洋三

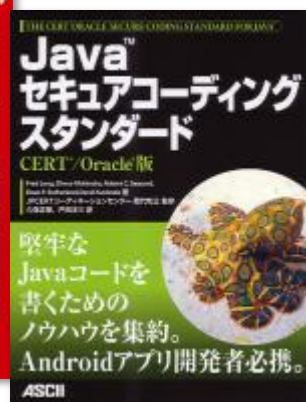
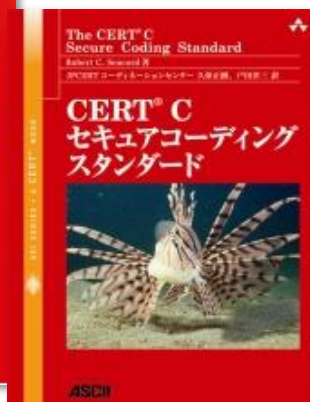
自己紹介

JPCERT/CC 情報流通対策グループ 解析チーム リードアナリスト 戸田 洋三



<http://www.tomo.gr.jp/root/e9706.html>

脆弱性情報分析, セキュアコーディング普及啓発活動.....
に努めてます



JPCERT/CCをご存知ですか？

JPCERT/CC について

- JPCERT/CC = “コーディネーションセンター”
- インシデントの解決・再発防止に向けて国内外のCSIRTや関係機関と**調整を行う**組織
- 国際連携が必要なオペレーションや情報連携に関する、**日本の窓口となるCSIRT**(窓口CSIRT)

※各国に同様の窓口となるCSIRTが存在する

(例えば、米国のUS-CERT,CERT/CC、中国のCNCERT、韓国のKrCERT/CC、など)

JPCERT/CCについて

インシデント予防

インシデントの予測と捕捉

発生したインシデントへの対応

脆弱性情報ハンドリング

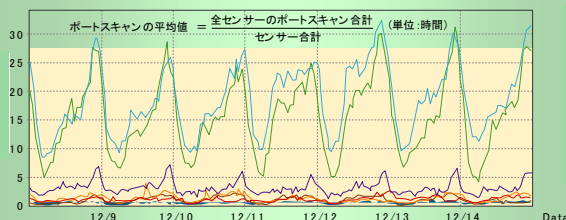
- 未公開の脆弱性関連情報を製品開発者へ提供し、対応依頼
- 関係機関と連携し、国際的に情報公開日を調整
- セキュアなコーディング手法の普及
- 制御システムに関する脆弱性関連情報の適切な流通



JVN Japan Vulnerability Notes

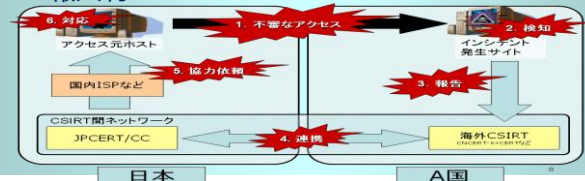
情報収集・分析・発信 定点観測 (TSUBAME)

- ネットワークトラフィック情報の収集分析
- セキュリティ上の脅威情報の収集、分析、必要とする組織への提供



インシデントハンドリング (インシデント対応調整支援)

- マルウェアの接続先等の攻撃関連サイト等の閉鎖等による被害最小化
- 攻撃手法の分析支援による被害可能性の確認、拡散抑止
- 再発防止に向けた関係各関の情報交換及び情報共有



早期警戒情報

重要インフラ、重要情報インフラ事業者等の特定組織向け情報発信

CSIRT構築支援

海外のNational-CSIRTや企業内のセキュリティ対応組織の構築・運用支援

制御システムセキュリティ

制御システムに関するインシデントハンドリング、情報収集・分析発信

アーティファクト分析

マルウェア (不正プログラム) 等の攻撃手法の分析、解析

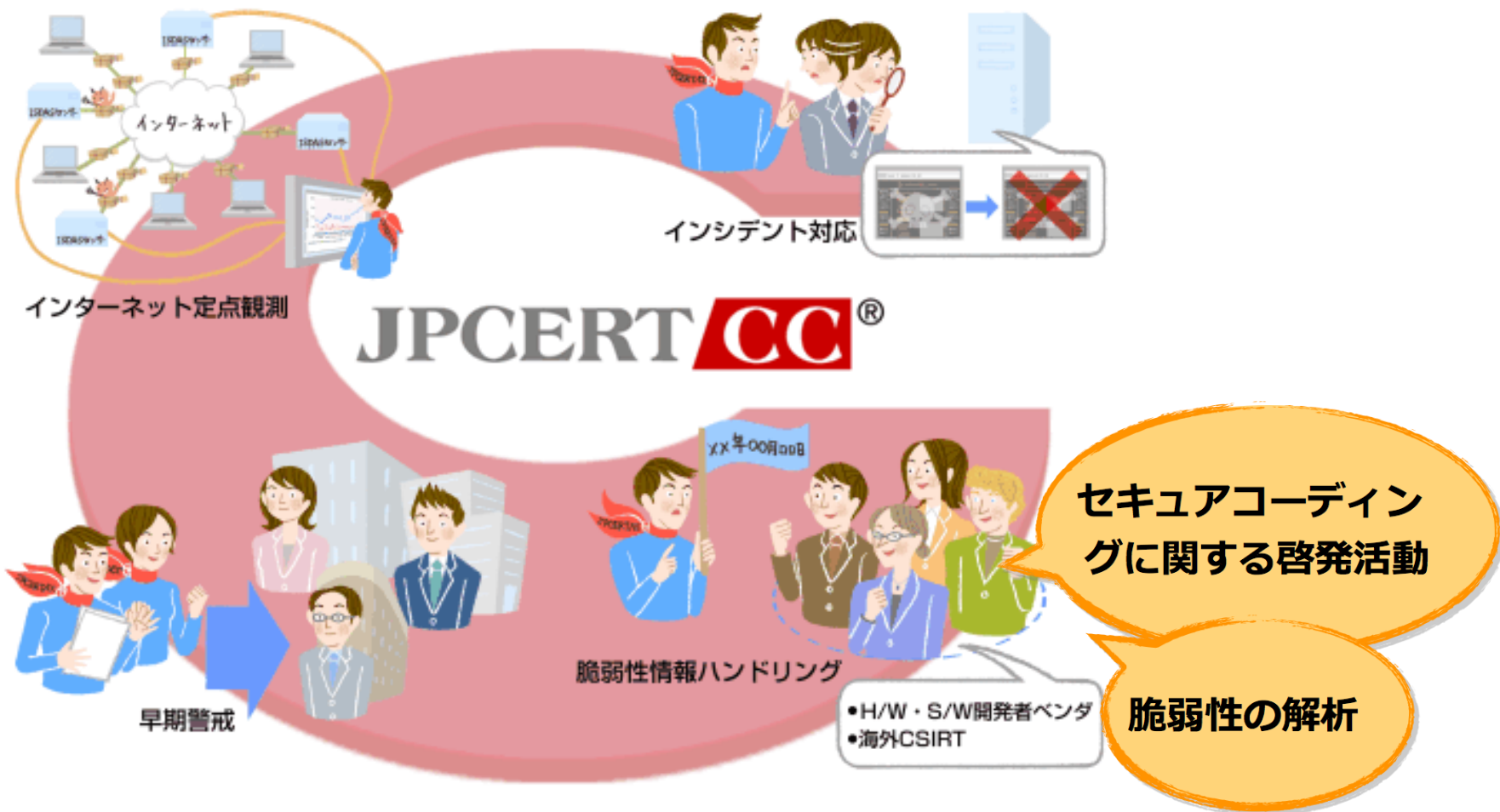
国内外関係者との連携

日本シーサート協議会、フィッシング対策協議会の事務局運営等

国際連携

各種業務を円滑に行うための海外関係機関との連携

JPCERT/CCについて



本日の話題

- ✓ SSL/TLS証明書検証って？
- ✓ 証明書検証に関連する脆弱性事例紹介
- ✓ 脆弱性事例のポイントおさらい
- ✓ 参考情報など



SSL/TLS証明書検証って？

脆弱性事例

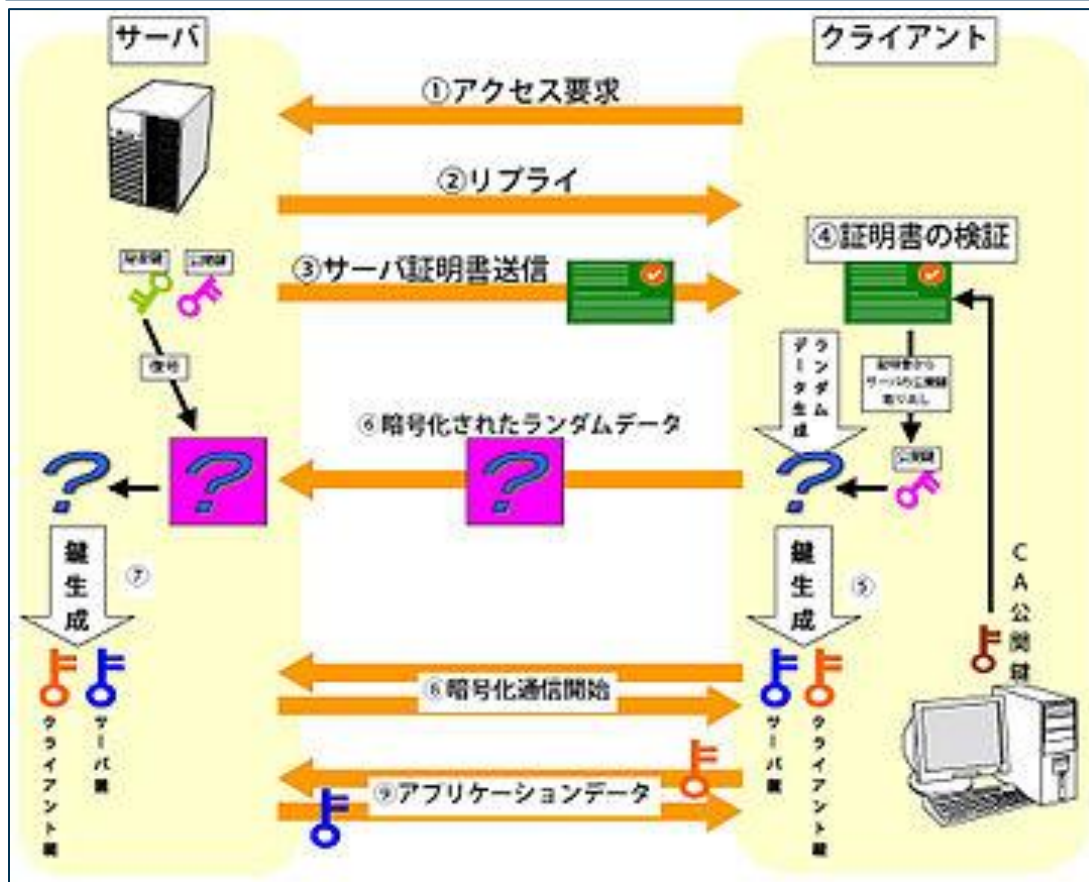
脆弱性事例ポイントおさらい

参考情報など

SSL/TLS通信手順

SSL/TLS

Transport層の上で暗号化通信路を実現する仕組み



SSL 3.0 - RFC6101
TLS 1.0 - RFC2246
TLS 1.1 - RFC4346
TLS 1.2 - RFC5246
.....

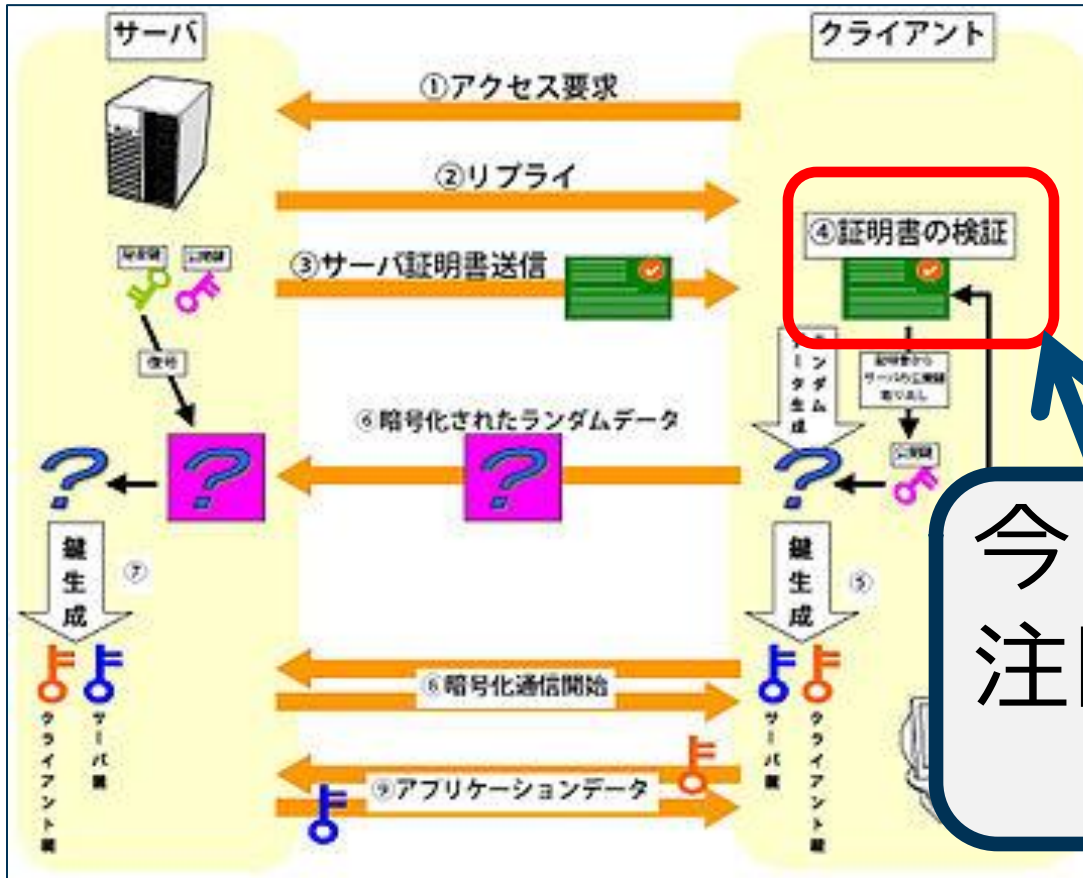
攻撃手法への対策や
新たな暗号化スイート追加など、現在も進化中...

参考: http://ja.wikipedia.org/wiki/Transport_Layer_Security

SSL/TLS通信手順

SSL/TLS

Transport層の上で暗号化通信路を実現する仕組み



SSL 3.0 - RFC6101
TLS 1.0 - RFC2246
TLS 1.1 - RFC4346
TLS 1.2 - RFC5246
.....

今日はここだけに
注目します。

参考: http://ja.wikipedia.org/wiki/Transport_Layer_Security

サーバ証明書って？

- サーバの公開鍵とサーバのドメイン名がはいつている (https で使う場合)
- とある CA (認証局)が, ドメイン名と公開鍵の関係を確認してくれたもの



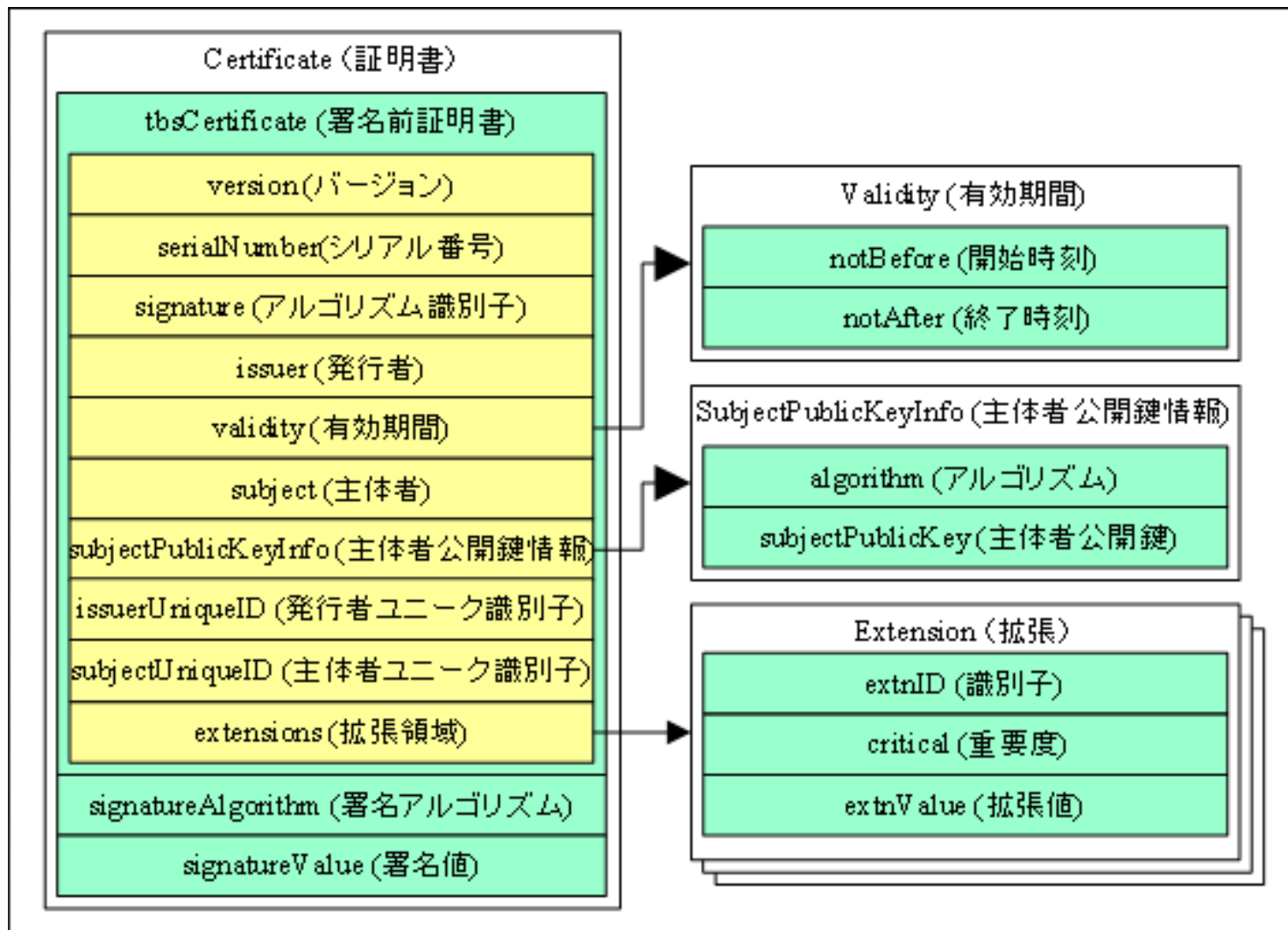
- ITU-T規格 X.509 で決められているものを流用
 - RFC5280, RFC6818
- Web ブラウザなどでは, 信頼する CA のオレオレ証明書をあらかじめ持っておく



証明書の検証には2つの処理が含まれている

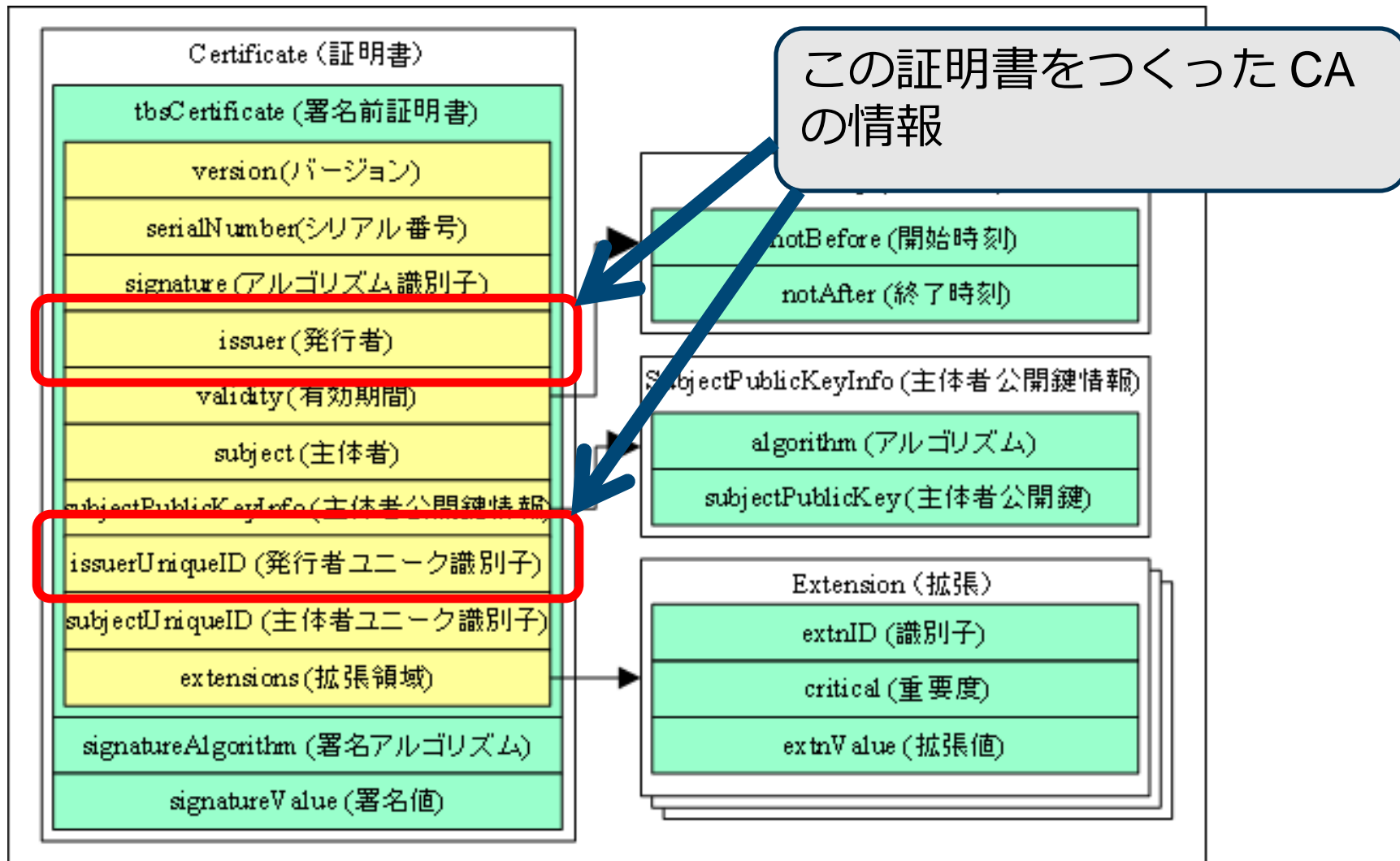
- 受け取ったサーバ証明書が、信頼する CA を起点として正しく作られたものであることを確認する
 - ⇒**証明書検証**
- 信頼できるサーバ証明書が持っているサーバ名がアクセスしようとしているサーバ名と一致することを確認する
 - ⇒**ホスト名検証**

X.509 v3 証明書の構造



<https://www.ipa.go.jp/security/pki/033.html>

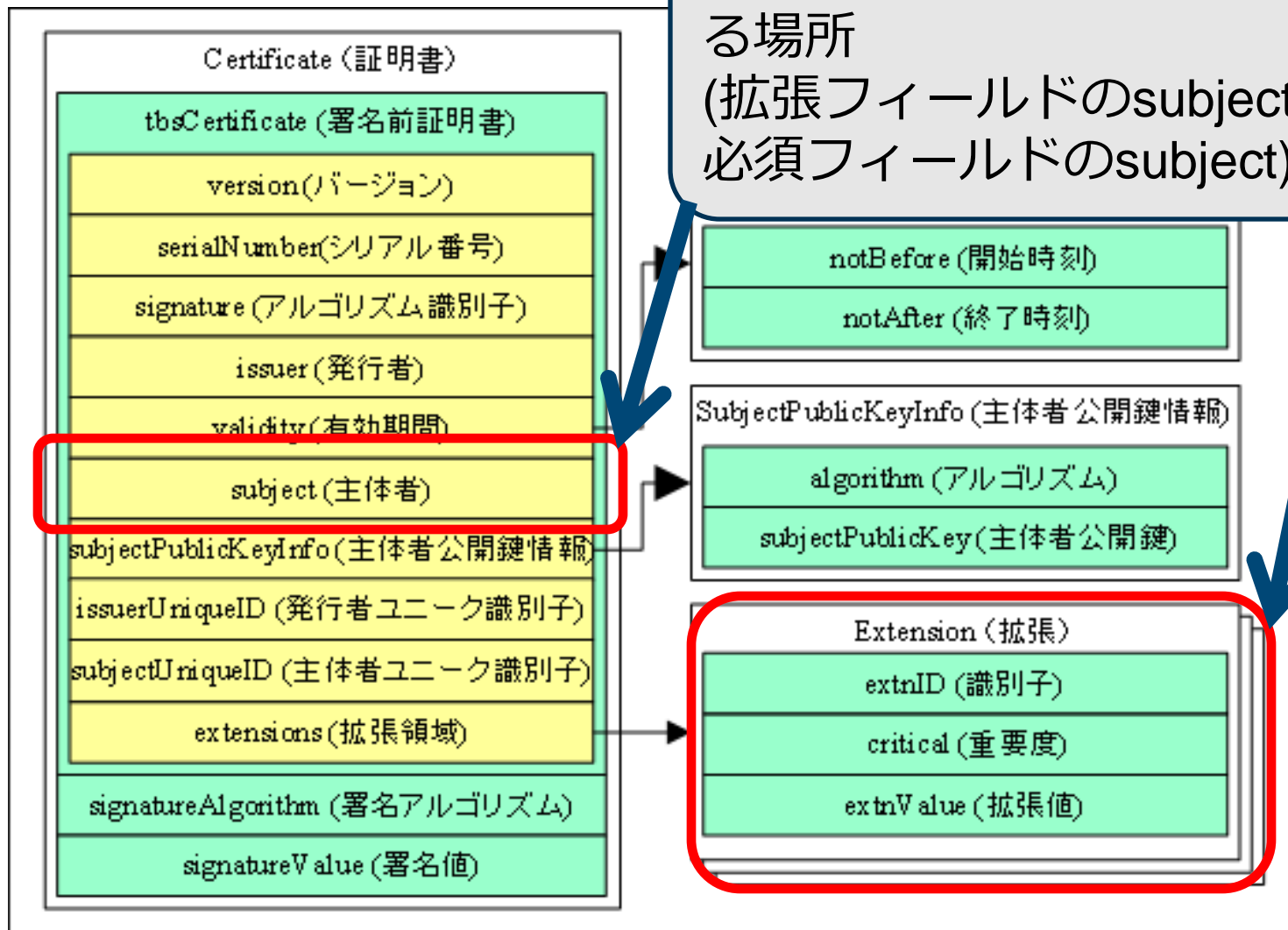
X.509 v3 証明書の構造



<https://www.ipa.go.jp/security/pki/033.html>

X.509 v3 証明書の構造

サーバのドメイン名が収められている場所
(拡張フィールドのsubjectAltNameと必須フィールドのsubject)



<https://www.ipa.go.jp/security/pki/033.html>

X.509 証明書検証

- ASN.1 データ構造として正しく読めるか?
- CAによる署名は正しくなされているか?
- 有効期限は切れていないか?
- 失効されていないか?
- 証明書チェーンをたどって信頼している(CAの)証明書に辿りつけるか?**

RFC5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

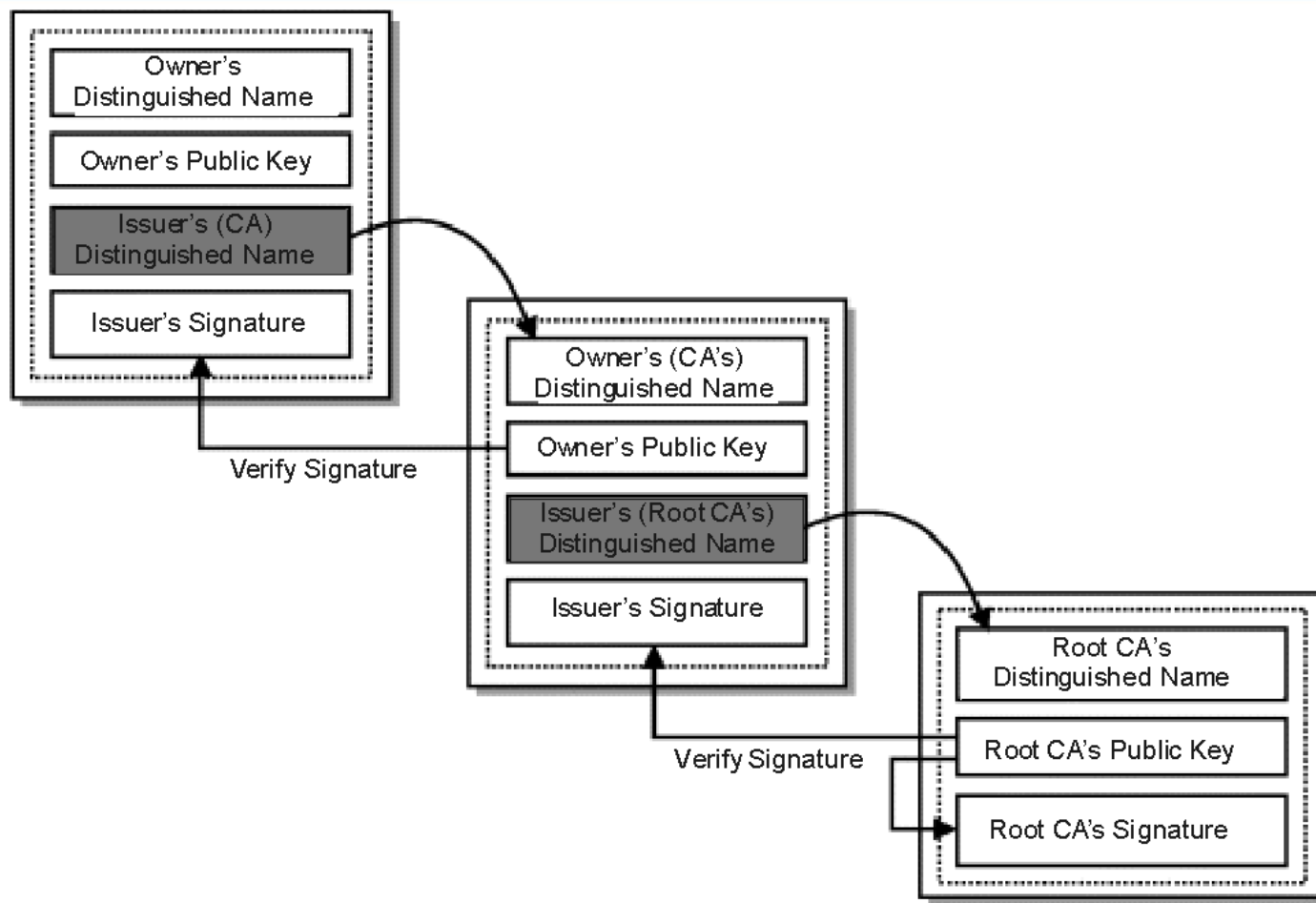
6. Certification Path Validation

<http://tools.ietf.org/html/rfc5280#section-6>

Certification path validation algorithm

http://en.wikipedia.org/wiki/Certification_path_validation_algorithm

証明書チェーンの検証処理



<https://security.stackexchange.com/questions/56389/ssl-certificate-framework-101-how-does-the-browser-actually-verify-the-validity>

ホスト名検証

- アクセスしようとしているサーバ名(ドメイン名)とサーバ証明書に示されているサーバ名が一致するか?
- サーバ証明書に subjectAltName が含まれている場合はそれを使うこと
- 一致するかどうかの検証アルゴリズムは証明書検証で使うアルゴリズムと同じ

RFC2818: HTTP Over TLS

3.1. Server Identity

<http://tools.ietf.org/html/rfc2818#section-3.1>

RFC5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

7. Processing Rules for Internationalized Names

<http://tools.ietf.org/html/rfc5280#section-7>

SSL/TLS証明書検証って？

脆弱性事例

脆弱性事例ポイントおさらい

参考情報など

証明書を検証不備

jvn.jp の掲載事例で「証明書の検証不備」を拾い出してみると... (2013年以降)

JVN#27388160: Android 版「スマ保」における SSL/TLS サーバ証明書の検証不備の脆弱性
JVN#48270605: Yahoo!ボックス(Android版)における SSL サーバ証明書の検証不備の脆弱性
JVN#04560253: Android 版アプリ「ゆこゆこ」における SSL サーバ証明書の検証不備の脆弱性
JVN#17637243: Android 版アプリ Kindle における SSL サーバ証明書の検証不備の脆弱性
JVN#27702217: Android 版 Ameba における SSL サーバ証明書の検証不備の脆弱性
JVN#72950786: Android 版 Outlook.com における SSL サーバ証明書の検証不備の脆弱性
JVN#10603428: Android 版アプリ「JR東日本アプリ」における SSL サーバ証明書の検証不備の脆弱性
JVN#16263849: Android 版アプリ「出前館」における SSL サーバ証明書の検証不備の脆弱性
JVN#48810179: Android 版アプリ「デニーズ」における SSL サーバ証明書の検証不備の脆弱性
JVN#97810280: KDrive個人版 PCクライアントソフトにおける SSL サーバ証明書の検証不備の脆弱性
JVN#75084836: Android 版 Yahoo!ショッピングにおける SSL サーバ証明書の検証不備の脆弱性
JVN#68156832: ヤフオク! における SSL サーバ証明書の検証不備の脆弱性
JVN#39218538: Android 版 ピザハット公式アプリ 宅配ピザのPizzaHut における SSL サーバ証明書の検証不備の脆弱性
JVN#85812843: FileMaker Pro における SSL サーバ証明書の検証不備の脆弱性
JVN#VU#389795: Windows Phone 7 に SSL サーバ証明書の検証不備の脆弱性
JVN#39707339: Opera における SSL サーバ証明書の検証不備の脆弱性
JVN#82029095: spモードメールアプリにおける SSL サーバ証明書の検証不備の脆弱性

Android アプリでよく報告されています。
(もちろん他の実行環境の例もあり)

参考

SSL/TLS ではなく IPsec での例もあり: CVE-2013-1028
Apple OS X v10.8.5 and Security Update 2013-004

某宅配ピザアプリにおけるSSL証明書検証不備の脆弱性

公開日：2013/06/07 最終更新日：2013/06/07

JVN#39218538

Android 版 ピザハット公式アプリ 宅配ピザのPizzaHut における SSL サーバ証明書の検証不備の脆弱性

概要

Android 版 ピザハット公式アプリ 宅配ピザのPizzaHut には、SSL サーバ証明書の検証不備の脆弱性が存在します。

影響を受けるシステム

- Android 版 宅配ピザのPizzaHut 1.1.0.a およびそれ以前

開発者によると、iOS 版では本脆弱性の影響を受けないとのこと。

詳細情報

Android 版 ピザハット公式アプリ 宅配ピザのPizzaHut には、SSL サーバ証明書の検証

想定される影響

中間者攻撃 (man-in-the-middle attack) による暗号通信の盗聴などが行なわ

対策方法

アップデートする

開発者が提供する情報をもとに、最新版へアップデートして

ベンダ情報

ベンダ

日本ケンタッキー・フライド・チキン株式会社 (以下「ケンタッキ

<https://jvn.jp/jvn/39218538/index.html>

<https://play.google.com/store/apps/details?id=jp.pizzahut.aorder>

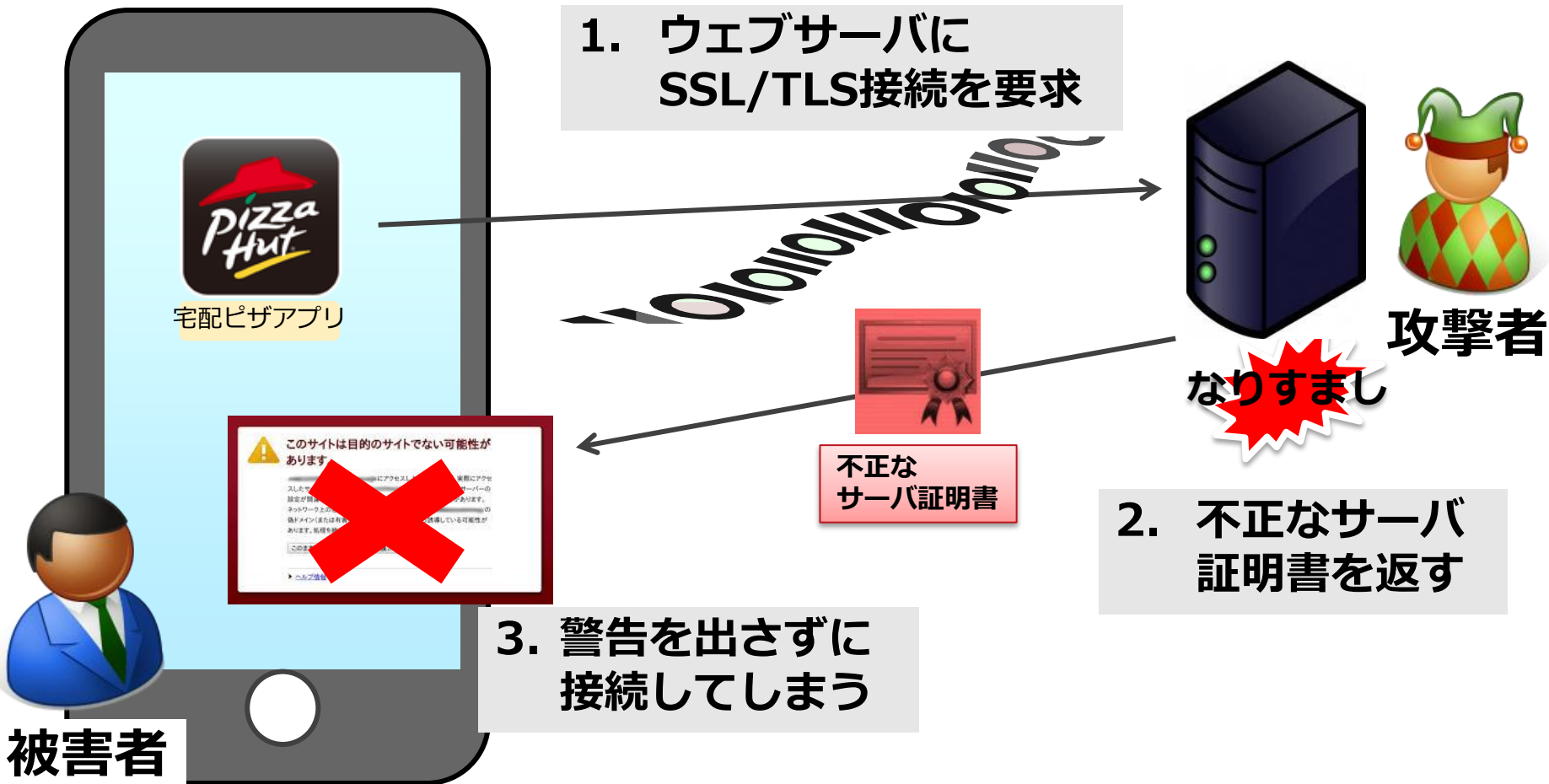
通信の盗聴や改ざん
アプリユーザー
の個人情報

宅配ピザのデリバリーを予
めしたり注文したりできる
アプリ。

通信にはユーザの
氏名、住所、メー
ルアドレスなどの
情報が含まれる

修正済み

攻撃のシナリオ



脆弱性の原因

jp/pizzahut/aorder/data/DataUtil.java

```
public static HttpClient getNewHttpClient() {
    DefaultHttpClient v6;
    try {
        KeyStore v5 = KeyStore.getInstance(KeyStore.getDefaultType());
        v5.load(null, null);
        MySSLSocketFactory mySSLScoket = new MySSLSocketFactory(v5);
        if(PizzaHutDefineRelease.sAllowAllSSL) {
            ((SSLSocketFactory)mySSLScoket).setHostnameVerifier
                (SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);
        }

        BasicHttpParams v2 = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(((HttpParams)v2), 30000);
        ...
    }
    catch(Exception v1) {
        v6 = new DefaultHttpClient();
    }
    return ((HttpClient)v6);
}
```

脆弱性の原因

```
public static HttpClient getNewHttpClient() {
    DefaultHttpClient v6;
    try {
        KeyStore v5 = KeyStore.getInstance("PKCS12");
        v5.load(null, null);
        MySSLConnectionFactory mySSLConnectionFactory =
            if(PizzaHutDefineRelease.sAllowAllHostnames)
                ((SSLConnectionFactory)mySSLConnectionFactory).setHostnameVerifier
                    (SSLConnectionFactory.ALLOW_ALL_HOSTNAME_VERIFIER);
    }

    BasicHttpParams v2 = new BasicHttpParams();
    HttpConnectionParams.setConnectionTimeout((HttpConnectionParams)v2), 30000);
    ...
}
catch(Exception v1) {
    v6 = new DefaultHttpClient();
}
return ((HttpClient)v6);
}
```

public class Summary: Ctors | Methods | Inherited Methods | [Expand All]
Added in API level 1

AllowAllHostnameVerifier

extends [AbstractVerifier](#)

[java.lang.Object](#)
↳ [org.apache.http.conn.ssl.AbstractVerifier](#)
↳ [org.apache.http.conn.ssl.AllowAllHostnameVerifier](#)

Class Overview

The ALLOW_ALL HostnameVerifier essentially turns hostname verification off. This implementation is a no-op, and never throws the SSLException.

ホスト名検証を無効にしている!

他の脆弱なパターン

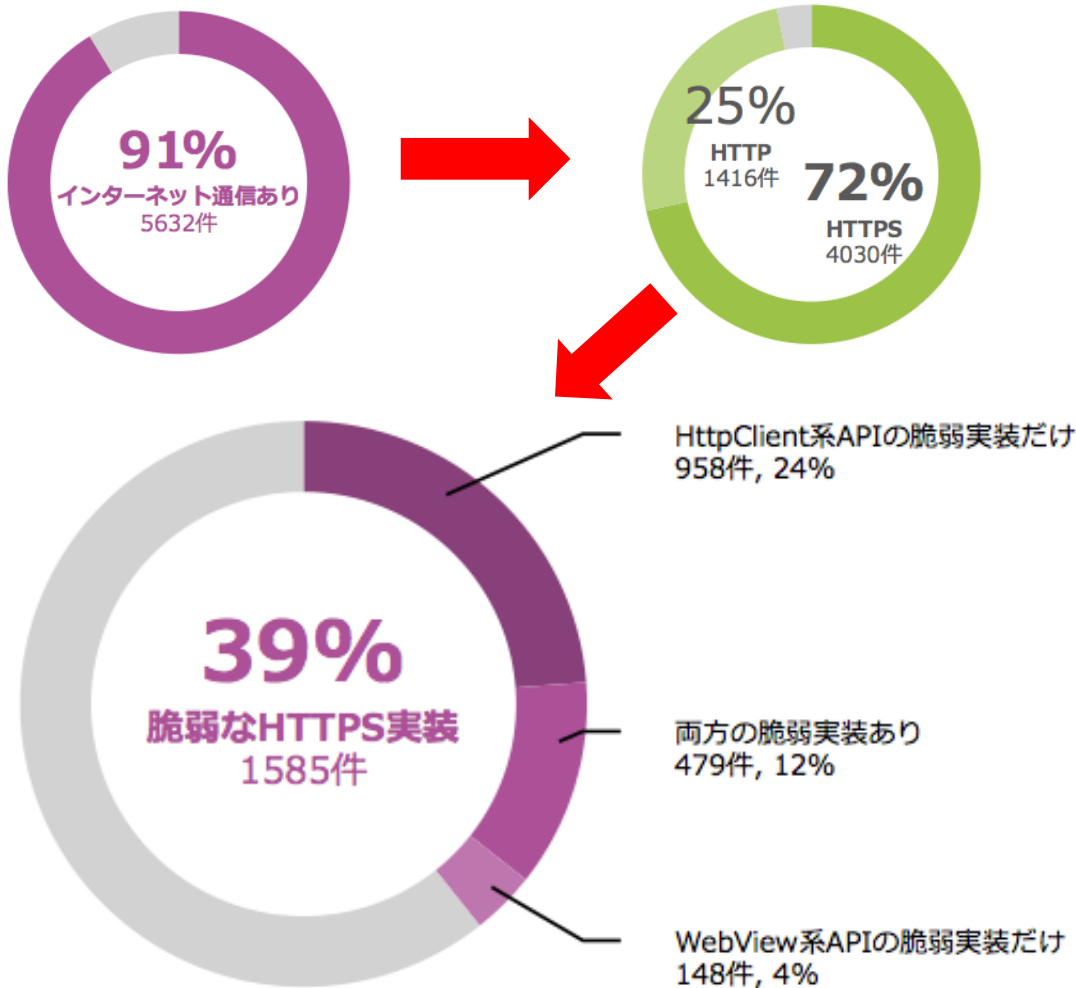
空っぽの HostnameVerifier

```
HostnameVerifier hv = new HostnameVerifier() {  
    @Override  
    public boolean verify(String hostname, SSLSession session) {  
        // 常に true を返す → どんなホスト名でも受付ける  
        return true;  
    }  
};
```

空っぽの TrustManager

```
TrustManager tm = new X509TrustManager() {  
    @Override  
    public void checkClientTrusted(X509Certificate[] chain,  
        String authType) throws CertificateException {  
        // 何もしない → どんな証明書でも受付ける  
    }  
    @Override  
    public void checkServerTrusted(X509Certificate[] chain,  
        String authType) throws CertificateException {  
        // 何もしない → どんな証明書でも受付ける  
    }  
    @Override  
    public X509Certificate[] getAcceptedIssuers() {  
        return null;  
    }  
};
```

Androidアプリの25%は脆弱なHTTPSを実装



アプリの4つ
に1つは脆弱
なHTTPS実装
を抱えている

Androidアプリ脆弱性調査レポート 2013年10月版
http://www.sonydna.com/sdna/solution/android_vulnerability_report_201310.pdf

脆弱なHTTPS実装の原因

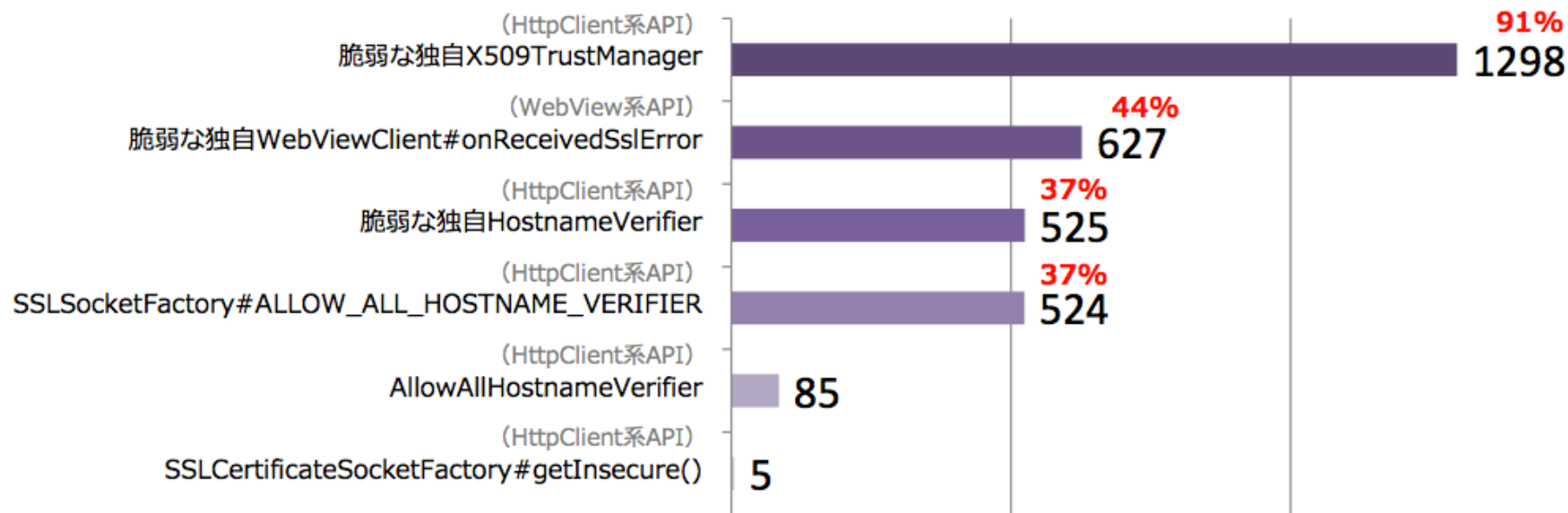


図 8 脆弱な HTTPS 実装の原因

Androidアプリ脆弱性調査レポート 2013年10月版
http://www.sonydna.com/sdna/solution/android_vulnerability_report_201310.pdf

ホスト名検証不備: CVE-2013-4073 Ruby

OpenSSL クライアントにおけるホスト名検証バイパス脆弱性 (CVE-2013-4073)

<https://www.ruby-lang.org/ja/news/2013/06/27/hostname-check-bypassing-vulnerability-in-openssl-client-cve-2013-4073/>



Ruby

A PROGRAMMER'S BEST FRIEND

“Rubyの SSL クライアントはホスト名前が一意であることをチェックする機構を持っていますが、その機構は **null バイト文字が含まれている** 証明書のホスト名前を適切に処理することができません。”

ホスト名検証不備: CVE-2014-3577 Apache HttpComponents

CVE-2014-3577 Apache HttpComponents client:
Hostname verification susceptible to MITM attack
<http://seclists.org/fulldisclosure/2014/Aug/48>

類似事例: CVE-2012-6153, CVE-2012-5783

“Apache HttpComponents ... may be susceptible to a 'Man in the Middle Attack' due to **a flaw in the default hostname verification** during SSL/TLS when a **specially crafted** server side certificate is used.”

“a (crafted) DN with a O field such as
O="foo,CN=www.apache.org"
and ordered such that the O appears prior to the CN field would
incorrectly match on the <www.apache.org> ...”

証明書チェーン検証不備: Fake ID

Android Fake ID Vulnerability Lets Malware Impersonate Trusted Applications, Puts All Android Users Since January 2010 At Risk

<https://bluebox.com/technical/android-fake-id-vulnerability/>



Presented at BlackHat 2014 USA

ANDROID FAKEID VULNERABILITY WALKTHROUGH

<https://www.blackhat.com/us-14/archives.html#android-fakeid-vulnerability-walkthrough>

Android OS のアプリ署名に関する脆弱性

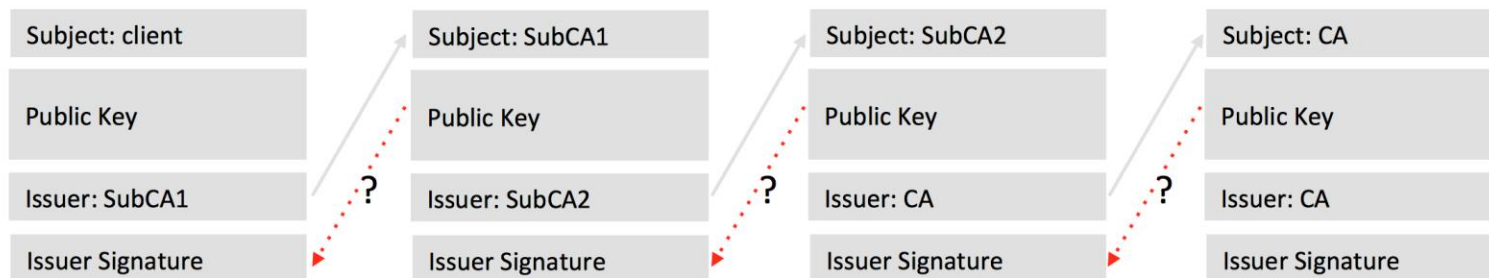
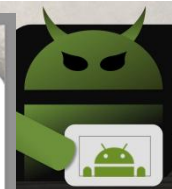
証明書チェーン検証不備: Fake ID



- Android アプリには電子署名がつけられている
- Android OS はアプリインストール時に電子署名を確認
 - SSL/TLSにおけるサーバ証明書検証と同等の処理
- 電子署名を確認するコードは Apache Harmony 由来のもの
- 証明書チェーンの確認に問題あり

Subject/Issuer の文字列としての一致のみを確認し、署名の確認を行っていなかった!!

“there is a conspicuous **absence of cryptographic verification** of any issuer cert claims, instead defaulting to **simple subjectDN to issuerDN string matching.**”



A certificate can **claim** to be issued by any other certificate ...

... and that claim is **not verified**

PKI Chaining - Android

BlackHat2014における発表資料から

証明書チェーン検証不備: Apple iOS

TWSL2011-007: iOS SSL Implementation Does Not Validate Certificate Chain

<http://blog.spiderlabs.com/2011/07/twsl2011-007-ios-ssl-implementation-does-not-validate-certificate-chain.html>

<https://www3.trustwave.com/spiderlabs/advisories/TWSL2011-007.txt>

“iOS's SSL certificate parsing contains a flaw where it fails to check the **basicConstraints** parameter of certificates in the chain.”

証明書チェーンのなかに「CAではない」という属性の証明書があっても、エラーにしていなかった

- Why Eve and Mallory Love Android: An Analysis of Android SSL (In)Security

<http://www2.dcsec.uni-hannover.de/files/android/p50-fahl.pdf>

- The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software

<https://crypto.stanford.edu/~dabo/pubs/abstracts/ssl-client-bugs.html>

多くのアプリが SSL/TLS ライブラリを誤用している!!

- 証明書検証を無効にしてしまう
- ホスト名検証を無効にしてしまう
-

■ Rethinking SSL Development in an Appified World

<http://android-ssl.org/files/p49.pdf>

SSL/TLS関連の脆弱性の発生原因を探る

- 開発者の理解不足
- 開発中の一時的な設定のままリリース
- ユーザからの要望

SSL/TLS証明書検証って？

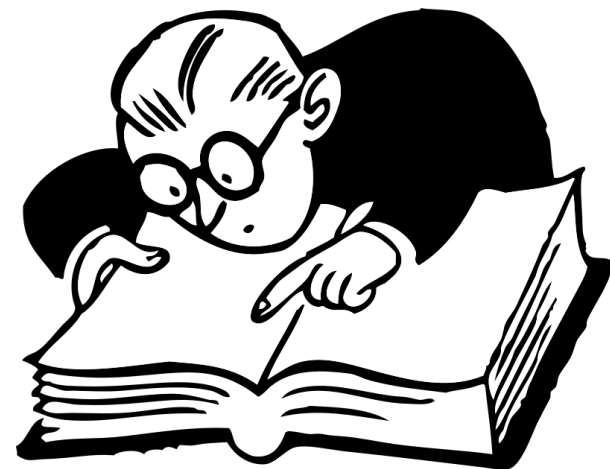
脆弱性事例

脆弱性事例ポイントおさらい

参考情報など

ポイント1: 証明書を検証する

- SSL/TLS 通信を行うならちゃんと証明書を検証しましょう
- 動作テストで検証を無効にしたのをそのまま公開、なんてダメよ
- Java/Androidの場合
 - SSLException は無視しない
 - TrustManager を独自実装しない
 - HostnameVerifier を独自実装しない



ポイント2: 証明書チェーンとホスト名検証に注意

- ライブラリなどで提供されている場合には下手にカスタマイズせずにそのまま使おう
- 自前で実装しなければならない場合は…
 - 仕様をしっかりと読み込む
 - 動作テストもしっかりとね
 - 既存の脆弱性事例を参考にして細工された証明書でも正しく動作するように注意する



参考: Proxy で動作検証

- Proxy 経由で通信させることで, 証明書の検証処理を確認できる
- 有名どころとしては Burp proxy, dsniff, Fiddler など
- これらのプロキシでは https 通信に対して自己署名証明書を提示したり, あらかじめ設定しておいた証明書を提示したりといった動作が可能
 - 信頼できない証明書(オレオレ証明書など)
 - ホスト名検証
 - 有効期限切れ
 - 失効済み証明書



SSL/TLS証明書検証って？

脆弱性事例

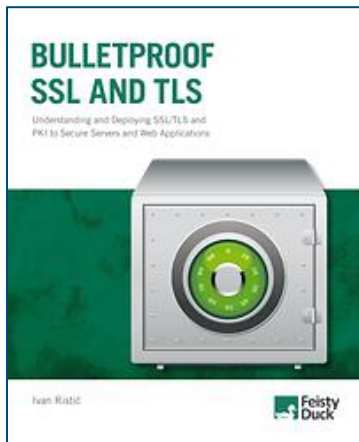
脆弱性事例ポイントおさらい

参考情報など



- マスタリングTCP/IP SSL/TLS編

- <http://shop.ohmsha.co.jp/shop/shopdetail.html?brandcode=000000001666&search=4-274-06542-1>



- Bulletproof SSL and TLS

- <https://www.feistyduck.com/books/bulletproof-ssl-and-tls/>

参考情報

●IPA: PKI 関連技術情報

- <https://www.ipa.go.jp/security/pki/index.html>

●Introduction to Public-Key Cryptography

- https://developer.mozilla.org/en-US/docs/Introduction_to_Public-Key_Cryptography

●Exciting Updates to Certificate Verification in Gecko

- <https://blog.mozilla.org/security/2014/04/24/exciting-updates-to-certificate-verification-in-gecko/>

●JNSA PKI Day

- PKI Day 2014 (<http://www.jnsa.org/seminar/pki-day/2014/>)

●一般社団法人日本スマートフォンセキュリティ協会 『Androidアプリのセキュア設計・セキュアコーディングガイド』

- <https://www.jssec.org/report/securecoding.html>

●SSL/TLSライブラリの正しい使い方(もしくは、コモンネームの検証について)

- <http://blog.kazuhooku.com/2014/01/sslts.html>

●Android Pinning by Moxie Marlinspike

- <https://github.com/moxie0/AndroidPinning>

References

- SSL Vulnerabilities: Who listens when Android applications talk?
 - <http://www.fireeye.com/blog/technical/2014/08/ssl-vulnerabilities-who-listens-when-android-applications-talk.html>
- Defeating SSL Certificate Validation for Android Applications
 - <https://secure.mcafee.com/us/resources/white-papers/wp-defeating-ssl-cert-validation.pdf>
- OnionKit by Android Library Project for Multi-Layer Network Connections (Better TLS/SSL and Tor)
 - <https://github.com/guardianproject/OnionKit>
- CERT/CC Vulnerability Note VU#582497: Multiple Android applications fail to properly validate SSL certificates
 - <https://www.kb.cert.org/vuls/id/582497>

付録

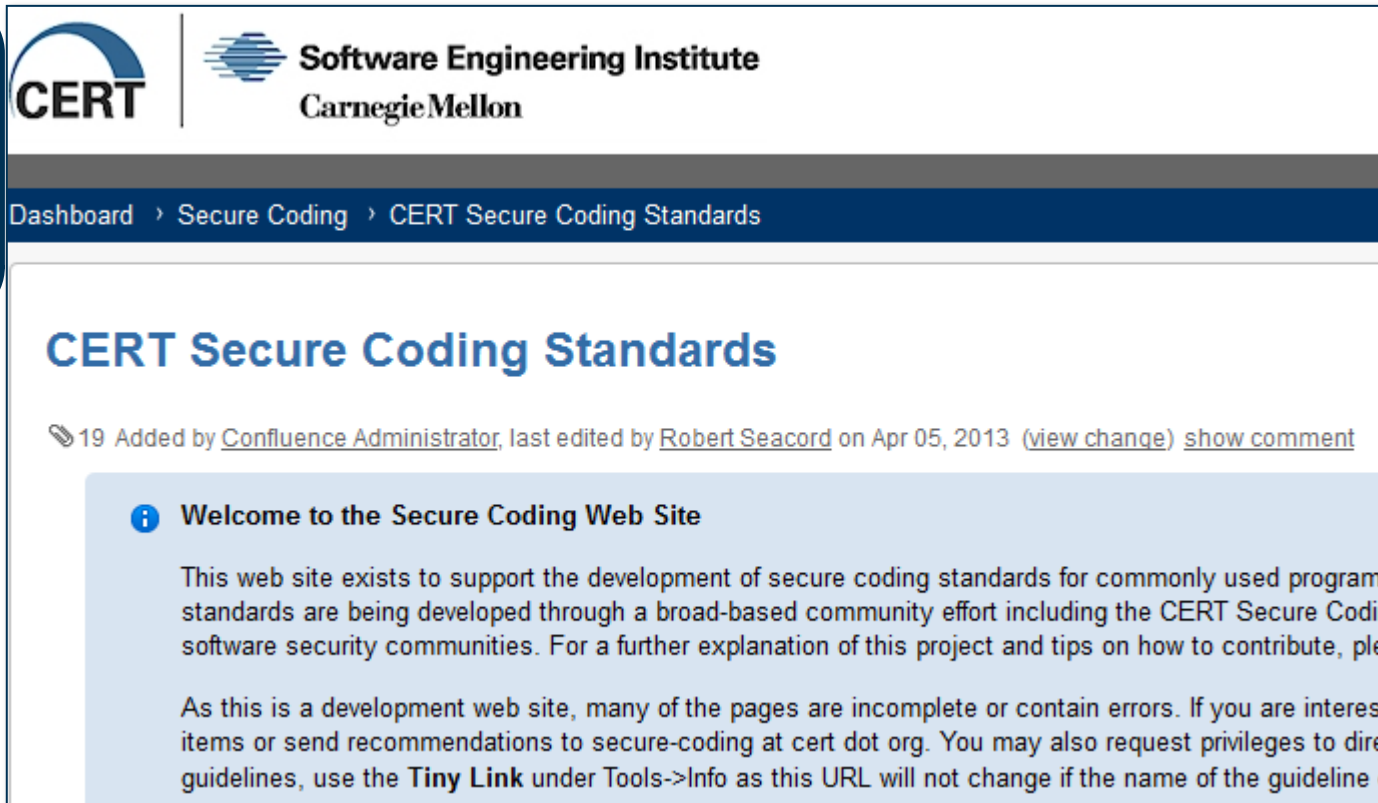
セキュアコーディングスタンダード



セキュアコーディングスタンダード

米国 CMU/SEI の the CERT Secure Coding Initiative
によるコーディングスタンダードシリーズ

<https://www.securecoding.cert.org/>

いまのところ、
4種類公開されて
います。



  **Software Engineering Institute**
Carnegie Mellon

Dashboard > Secure Coding > CERT Secure Coding Standards

CERT Secure Coding Standards

🔒 19 Added by [Confluence Administrator](#), last edited by [Robert Seacord](#) on Apr 05, 2013 ([view change](#)) [show comment](#)

📘 Welcome to the Secure Coding Web Site

This web site exists to support the development of secure coding standards for commonly used program standards are being developed through a broad-based community effort including the CERT Secure Coding software security communities. For a further explanation of this project and tips on how to contribute, please see the guidelines.

As this is a development web site, many of the pages are incomplete or contain errors. If you are interested in contributing items or send recommendations to [secure-coding at cert dot org](mailto:secure-coding@cert.org). You may also request privileges to direct the development of guidelines, use the **Tiny Link** under Tools->Info as this URL will not change if the name of the guideline changes.

C セキュアコーディングスタンダード

The CERT C Secure Coding Standard



Version 1.0 of The CERT C Secure Coding Standard is now available as a [book](#) from Addison-Wesley. This official release can be used as a fixed point of reference for the development of compliant applications and source code analysis tools.

Development of the next version of the [CERT C Secure Coding Standard](#) is being performed here on the secure coding wiki. This version is a work in progress and reflects the current thinking of the secure coding community. See [this link](#) for more information about the current thinking of the secure coding community.

There is a JPCERT/CC

JPCERT/CC で日本語公開中!!

<https://www.jpccert.or.jp/sc-rules/>

電子署名名: Japan Computer Emergency Response Team Coordination Center
DN: c=JP, st=Tokyo, l=Chiyoda-ku, email=office@jpccert.or.jp, o=Japan Computer Emergency Response Team Coordination Center, cn=Japan Computer Emergency Response Team Coordination Center
日付: 2011.03.29 19:30:54 +09'00'

Japan Computer Emergency Response Team Coordination Center

C/C++ セキュアコーディングセミナー
2010年度版

CERT C セキュアコーディングスタンダード

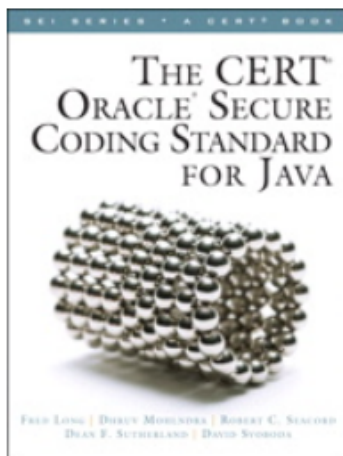
JPCERT コーディネーションセンター

CERT C セキュアコーディングスタンダード紹介

<https://www.jpccert.or.jp/research/materials.html#secure>

Java セキュアコーディングスタンダード

The CERT Oracle Secure Coding Standard for Java



Version 1.0 of *The CERT Oracle Secure Coding Standard for Java* is now available as a [book](#) from Addison-Wesley.

Development of the next version of the [The CERT Oracle Secure Coding Standard for Java](#) is being performed here on the secure coding wiki. This version is a work in progress and reflects the current thinking of the secure coding community. Subsequent official releases of this standard will be issued as dictated by the community.

Java is a truly global technology, used in many countries.

There is also a JPCERT/CC partner JPCERT/CC.

JPCERT/CC で日本語公開中!!

<https://www.jpccert.or.jp/java-rules/>

オープンソースの「今」を伝える
オープンソースカンファレンス
2011 Nagoya

オープンソースカンファレンス 2011 Nagoya
セキュアコーディング/ススめ
(Java編)

2011年08月20日
戸田 洋三

JPCERT コーディネーションセンター
secure-coding@jpccert.or.jp

Copyright © 2011 JPCERT/CC. All rights reserved. JPCERT/CC®

OSC2011@Nagoya でも紹介してます
http://www.ospn.jp/osc2011-nagoya/pdf/osc2011nagoya-JPCERT_CC.pdf

セキュアコーディングスタンダード(C++, Perl)

The CERT C++ Secure Coding Standard



The [CERT C++ Secure Coding Standard](#) is under development.
contribute new guidelines to this standard.

The CERT Perl Secure Coding Standard



The [CERT Perl Secure Coding Standard](#) is under development.

“under development”
(開発中)

お問合せ、インシデント対応のご依頼は

JPCERT コーディネーションセンター

— Email : office@jpcert.or.jp

— Tel : 03-3518-4600

— <https://www.jpcert.or.jp/>

インシデント報告

— Email : info@jpcert.or.jp

— <https://www.jpcert.or.jp/form/>

セキュアコーディング

— Email : secure-coding@jpcert.or.jp

— <https://www.jpcert.or.jp/securecoding>

- Home
- サイト内検索
- トップページ
- 情報提供
 - 注意喚起
 - 早期警戒
 - 脆弱性対策情報
 - Weekly Report
- 各種届出・申込
- 制御システムセキュリティ
- ラーニング
- 公開資料
 - 四半期レポート
 - 研究・調査レポート
 - CSIRTマテリアル
- イベント
- プレスリリース
- JPCERT/CC

- 関連組織
- FIRST
- JPCERT/CCはFIRSTのチームメンバーです。またJPCERT/CCスタッフがSteering CommitteeメンバーとしてFIRSTの運営に協力しています。
- APCERT
- JPCERT/CCはAPCERTの事務局

注意喚起

- 深刻な影響を及ぼす脆弱性に関する注意喚起
- 2009-06-10 [公開] 2009年6月 Microsoft セキュリティ情報(緊急1件)に関する注意喚起
- 2009-05-13 [公開] JavaScript が埋め込まれる Web サイトの改ざんに関する注意喚起
- 2009-06-13 [公開] Adobe Reader 脆弱性に関する注意喚起
- 2009-05-13 [公開] 2009年5月 Microsoft セキュリティ情報(緊急1件)に関する注意喚起
- 2009-04-15 [公開] 脆弱性に関する注意喚起

脆弱性関連

- ソフトウェアなどの脆弱性に関する注意喚起
- 2009-06-19 15:00 XOOOPS コア脆弱性に関する注意喚起
- 2009-06-19 14:32 A51 D.O.O. 製 activeCollab におけるクロスサイトスクリプティングの脆弱性
- 2009-06-19 14:32 Microsoft Word 2007におけるクロスサイトスクリプティングの脆弱性
- 2009-06-19 14:32 Serene におけるクロスサイトスクリプティングの脆弱性

Weekly Report

- 2009-06-17日

セキュリティインシデント...
フィッシングサイト...
Webサイトの改ざん...
マルウェア...
不正アクセス...

ISDAS
[インターネット定点観測]



インターネット上に配置したセンサーにより、セキュリティ上の脅威となるトラフィックを観測しています。

お薦めページ

セキュリティ対策講座



教育担当者が使える、新入社員などが身につけておくべきセキュリティ知識などを紹介しています。

イベント

- 第21回 FIRST Annual Conference 京都 参加申し込み受付中
- O/OH セキュアコーディング ハーフデイキャンプ参加申し込み