脆弱性発見者の目から見た、脆弱性対応の最前線

株式会社リクルートテクノロジーズ
サイバーセキュリティエンジニアリング部
西村 宗晃

西村 宗晃

株式会社リクルートテクノロジーズ
サイバーセキュリティエンジニアリング部
シニアセキュリティエンジニア

国内携帯電話メーカーでのセキュリティコンサルタントなどを経て2016年3月より現職。リクルートのID管理基盤のセキュリティ保守やリクルートグループ全社の脆弱性修正支援に携わる。趣味はブラウザの脆弱性を探すこと。2015年に報告した脆弱性は70件を超える。著書にブラウザハック（監訳）。主な講演歴にCODE BLUE 2015、AVTOKYO 2016、PacSec 2016。2014年よりセキュリティ・キャンプ全国大会講師

簡単に言うと

脆弱性が大好き

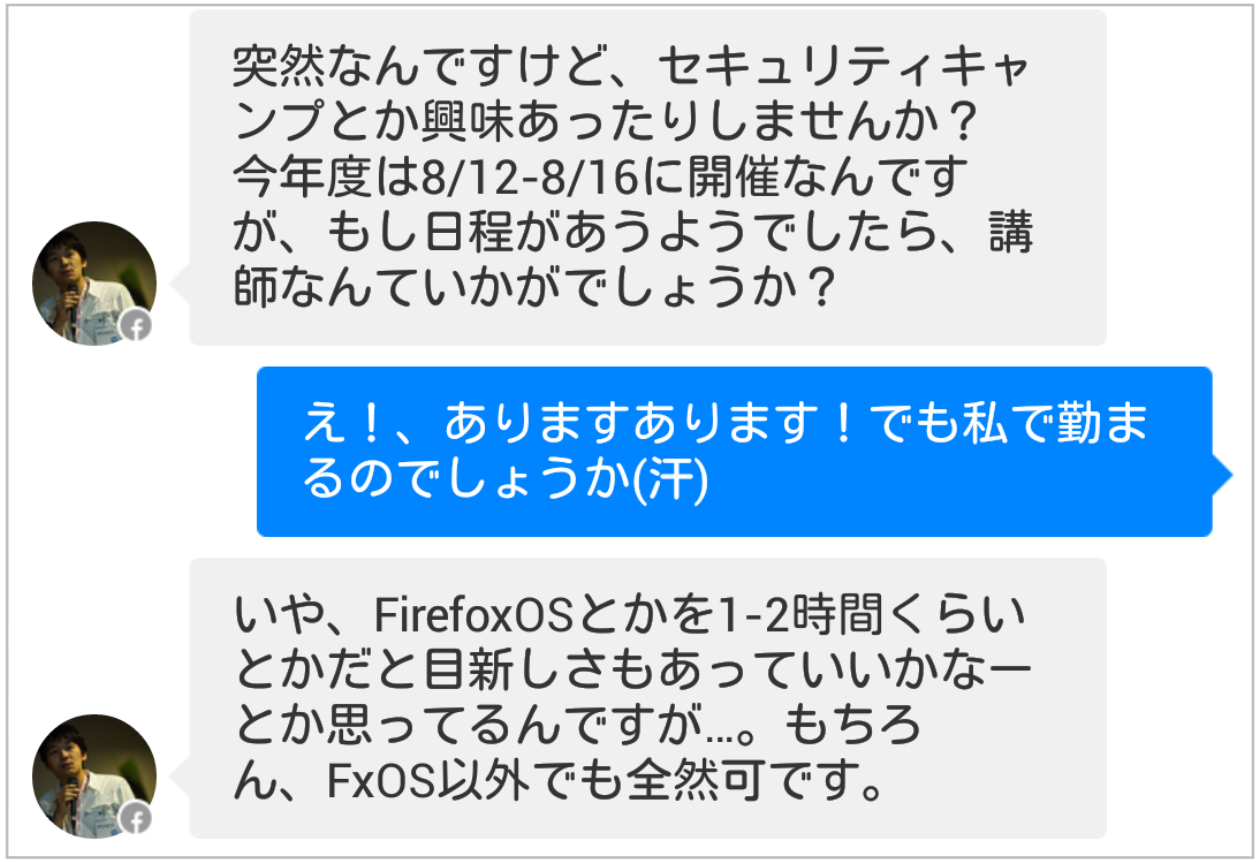
本日のお話

- 脆弱性を探し始めた理由と続ける理由
- 脆弱性を見つけるために実践していること
- 発見者の目から見た脆弱性対応の現場
- リクルートにおける脆弱性対応
- まとめ

脆弱性を探し始めた理由と続ける理由

きっかけは2014年のセキュリティ・キャンプ

- クラス長の長谷川陽介氏からフランクに**講師の依頼**が来た



セキュリティ・キャンプとは

- 次代を担うセキュリティ人材の発掘と育成を目的とした官民連携事業
  - IPAと35社の協賛企業からなる実施協議会が主催
  - 経済産業省が共催
  - 今年で13年目
- 4泊5日、朝から晩までセキュリティ漬けの合宿
  - 全国231名の応募者から選抜された51名が参加
  - 各領域の**一線で活躍する技術者陣が講師**を担当

安易に引き受けたものの・・・

- 自分以外は超有名な講師陣
  - 書店やネットのセキュリティ連載記事などでよく名前を見る人たち
  - Googleの脆弱性報告件数で世界2位の人
- CTFや未踏など多方面で活躍するチューター
- 全国から選び抜かれた応募者達
- 本当に**自分が講師で良かったのか**？
  - 自分の実力を客観的に測る手段が欲しかった
  - 教える資格があるかを確かめたかった

そこで、Firefoxブラウザの脆弱性報奨金制度へ挑戦

- 世界中のプレーヤーより先に脆弱性を見つけ出す競争
- 獲得した報奨金の額で実力を可視化できる
- 脆弱性を見つけられなければ、講師を辞退しようと考えていた

- 6ヶ月の苦難を経て**脆弱性を発見**！

脆弱性探しをして得たもの

- もちろん報奨金
  - これまでに1,000万円以上
  - でも、最初に手にした報奨金で自転車を買ったら、金銭欲が失せてしまった
- セキュリティエンジニアとしての**基礎力**
  - 難解な仕様やコードから目を背けずに読み解く力
  - 思い込みを捨て、物事を自分の手で検証する力

脆弱性を探し続ける理由

- 住宅ローンを返すため
  - 目標は40歳までに完済
  - 稼いだ実感のない報奨金は、借りた実感のないローンと相性がいい
- **技術力の維持向上**のため
  - 仕様やコードを悪用方法を考える習慣をつけておくと、
    業務で検査やレビューするときに攻撃方法が閃きやすくなる
  - 他の発見者や開発者と出会い、新たな攻撃の仕方を学ぶ

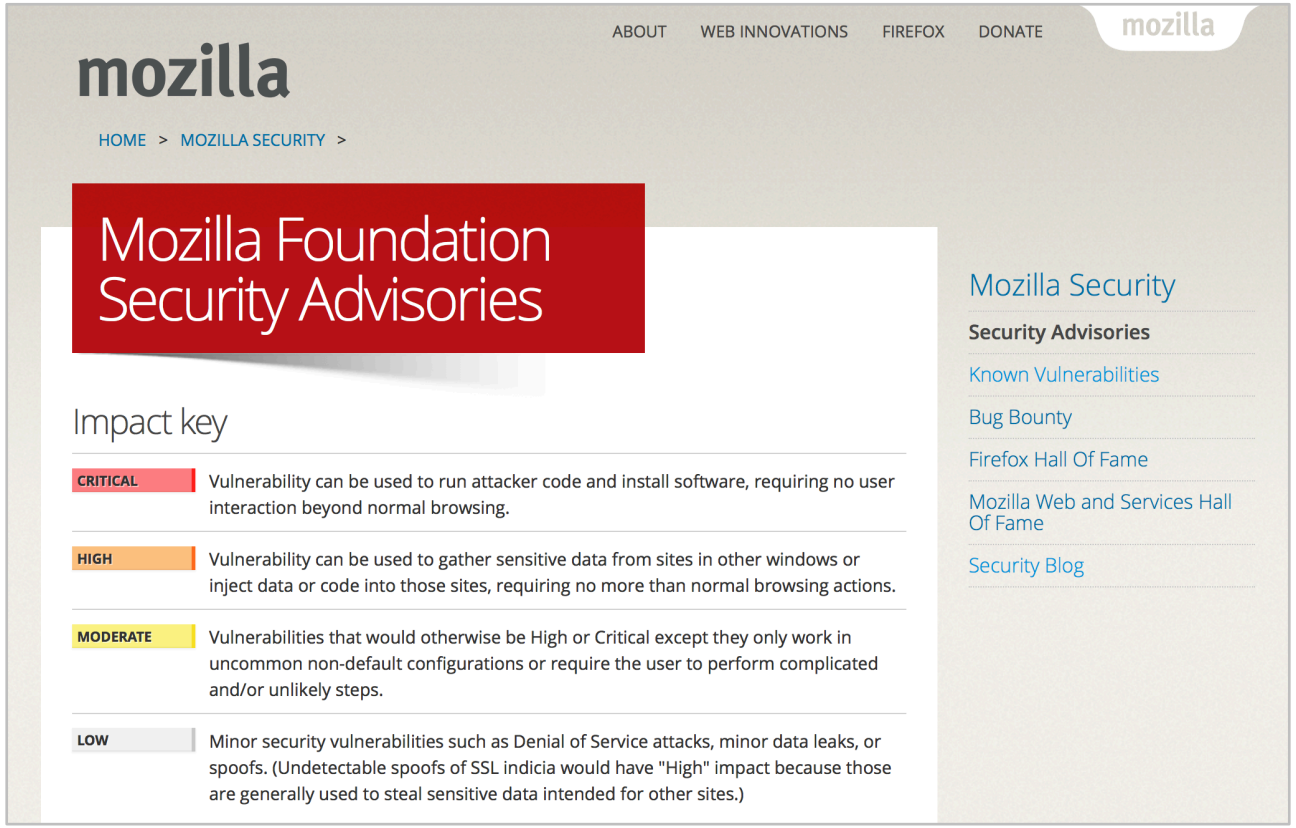
脆弱性を見つけるために実践していること

既知の脆弱性に学ぶ

- 過去の脆弱性情報を調べる
- 攻撃コードを作って**実際に検証**する
  - 改修の誤りなどが原因で、過去の脆弱性が再発する
  - 再利用できるように、作成した攻撃コードを保存しておく
- 類似の脆弱性を探す
  - 過去に脆弱性が指摘された機能や、それとよく似た機能
  - 前提条件を少し変えて、同じ攻撃コードを試す

既知の脆弱性に学ぶ - CSP違反レポート実装不備の事例

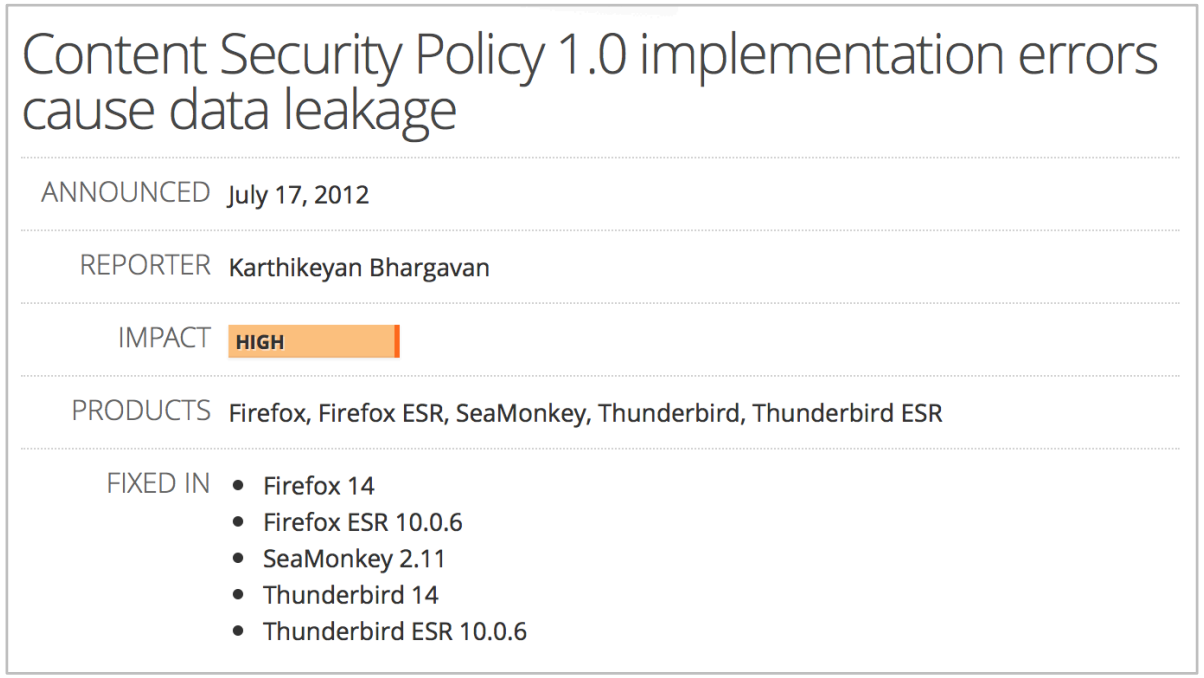
- Mozillaのセキュリティアドバイザリから過去の脆弱性情報を入手



https://www.mozilla.org/en-US/security/advisories/

既知の脆弱性に学ぶ - CSP違反レポート実装不備の事例

- 2012年7月、Content Security Policy（CSP）の違反レポート機能を通じて他のサイトの情報を盗めることをINRIAの研究者が指摘（CVE-2012-1963）



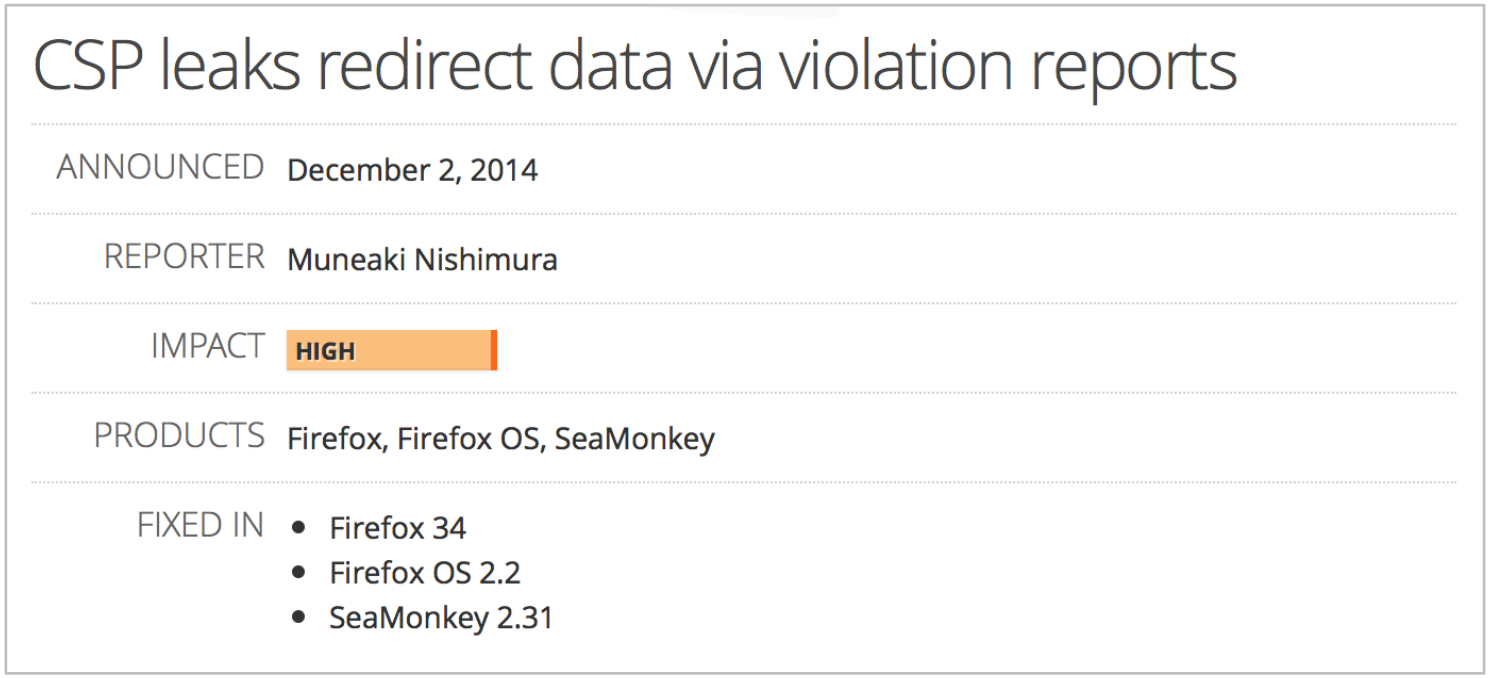
Content Security Policy 1.0 implementation errors cause data leakage

ANNOUNCED	July 17, 2012
REPORTER	Karthikeyan Bhargavan
IMPACT	HIGH
PRODUCTS	Firefox, Firefox ESR, SeaMonkey, Thunderbird, Thunderbird ESR
FIXED IN	• Firefox 14 • Firefox ESR 10.0.6 • SeaMonkey 2.11 • Thunderbird 14 • Thunderbird ESR 10.0.6

https://www.mozilla.org/en-US/security/advisories/mfsa2012-53/

既知の脆弱性に学ぶ - CSP違反レポート実装不備の事例

- 2014年12月、高速化のためにCSPの実装をリファクタリングした際、**同じ脆弱性が再発**（CVE-2014-1591）



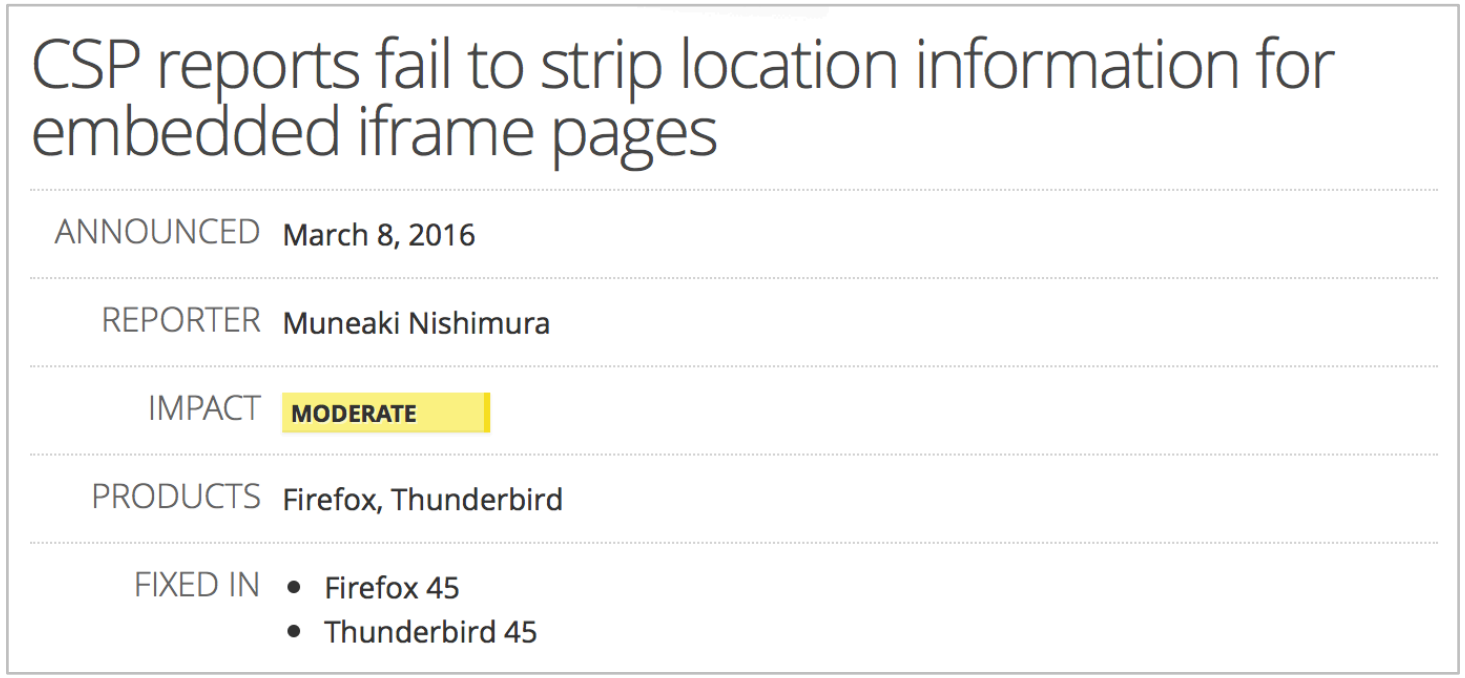
CSP leaks redirect data via violation reports

ANNOUNCED	December 2, 2014
REPORTER	Muneaki Nishimura
IMPACT	HIGH
PRODUCTS	Firefox, Firefox OS, SeaMonkey
FIXED IN	• Firefox 34 • Firefox OS 2.2 • SeaMonkey 2.31

https://www.mozilla.org/en-US/security/advisories/mfsa2014-86/

既知の脆弱性に学ぶ - CSP違反レポート実装不備の事例

- 2016年3月、iframe内のページでCSPに違反する処理を行うことにより**類似の脆弱性が再現**（CVE-2016-1955）



CSP reports fail to strip location information for embedded iframe pages

ANNOUNCED	March 8, 2016
REPORTER	Muneaki Nishimura
IMPACT	MODERATE
PRODUCTS	Firefox, Thunderbird
FIXED IN	• Firefox 45 • Thunderbird 45

https://www.mozilla.org/en-US/security/advisories/mfsa2016-18/

仕様に学ぶ

- IETFやW3Cの仕様書をひたすら読む
  - 複数の仕様をあわせ読むと、**仕様の抜け漏れ**が見えてくる
- 仕様書のSecurity Considerationsから攻撃の観点を得る
  - ブラウザの中で起きてはいけないことが分かる
- 機能が仕様どおりに実装されているか検証する
  - 仕様に書かれている**施策が実装されていない**ことがある

仕様に学ぶ – HTML Importsの実装不備の事例

- RFC 2183でContent-DispositionというHTTPヘッダが定義されている
  - Content-Disposition: attachmentがHTTPレスポンスに指定されている場合、ブラウザはそのコンテンツを開かず、ダウンロードさせなければならない
- **Content-Dispositionの無視は脆弱性**として扱われる
  - Firefox：CVE-2009-1306、CVE-2010-1197
  - Safari：CVE-2011-3426、CVE-2015-5921

仕様に学ぶ – HTML Importsの実装不備の事例

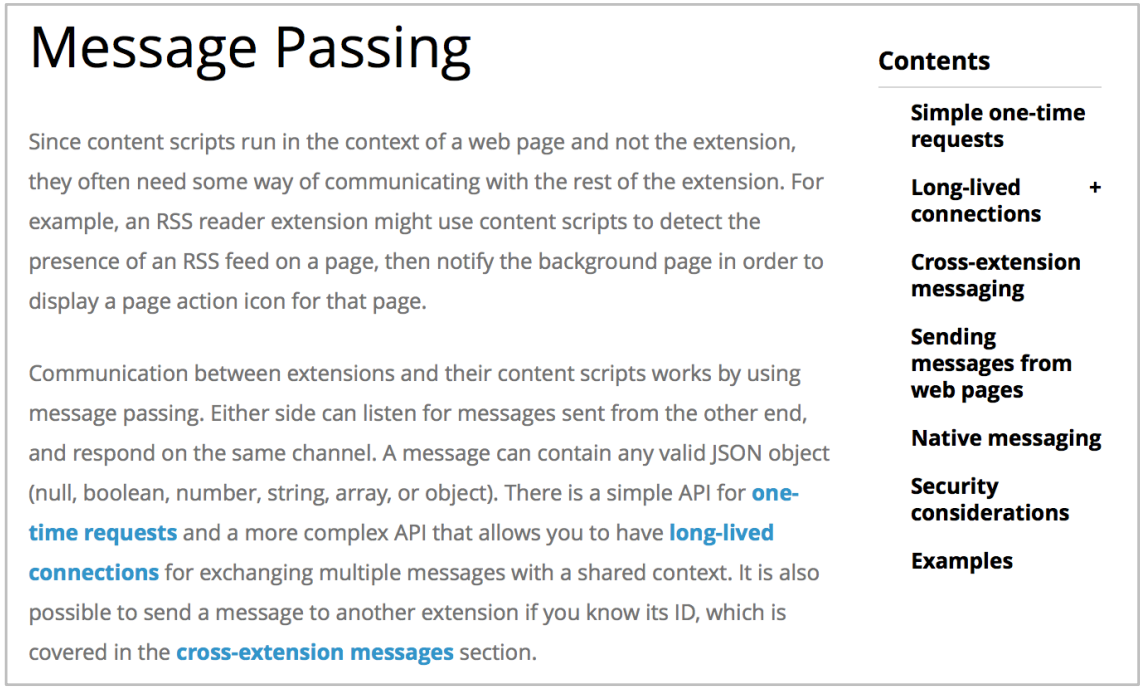
- ChromeのHTML ImportsがContent-Dispositionを無視することを開発元のGoogleに報告（Issue 439877）
- これを受け、**W3Cの仕様にContent-Dispositionの記述が追加**された

2. Fetch a resource from LOCATION with request's origin set to the origin of the master document, the mode to *CORS* and the credentials mode to *same-origin*.
    1. If fetched response type is *error* or the response has a header whose name is `Content-Disposition`:
        - Add LOCATION and **null** to the import map and **stop**.

https://www.w3.org/TR/2016/WD-html-imports-20160225/

仕様に学ぶ - Web Extensionsの実装不備の事例

- Chrome拡張機能のAPI仕様では、他の拡張機能との通信には Message Passingを使うことが求められている
- Message以外の手段で**他の拡張機能の動作に干渉**することは **禁止**されている（ようだ）



Message Passing

Since content scripts run in the context of a web page and not the extension, they often need some way of communicating with the rest of the extension. For example, an RSS reader extension might use content scripts to detect the presence of an RSS feed on a page, then notify the background page in order to display a page action icon for that page.

Communication between extensions and their content scripts works by using message passing. Either side can listen for messages sent from the other end, and respond on the same channel. A message can contain any valid JSON object (null, boolean, number, string, array, or object). There is a simple API for **one-time requests** and a more complex API that allows you to have **long-lived connections** for exchanging multiple messages with a shared context. It is also possible to send a message to another extension if you know its ID, which is covered in the **cross-extension messages** section.

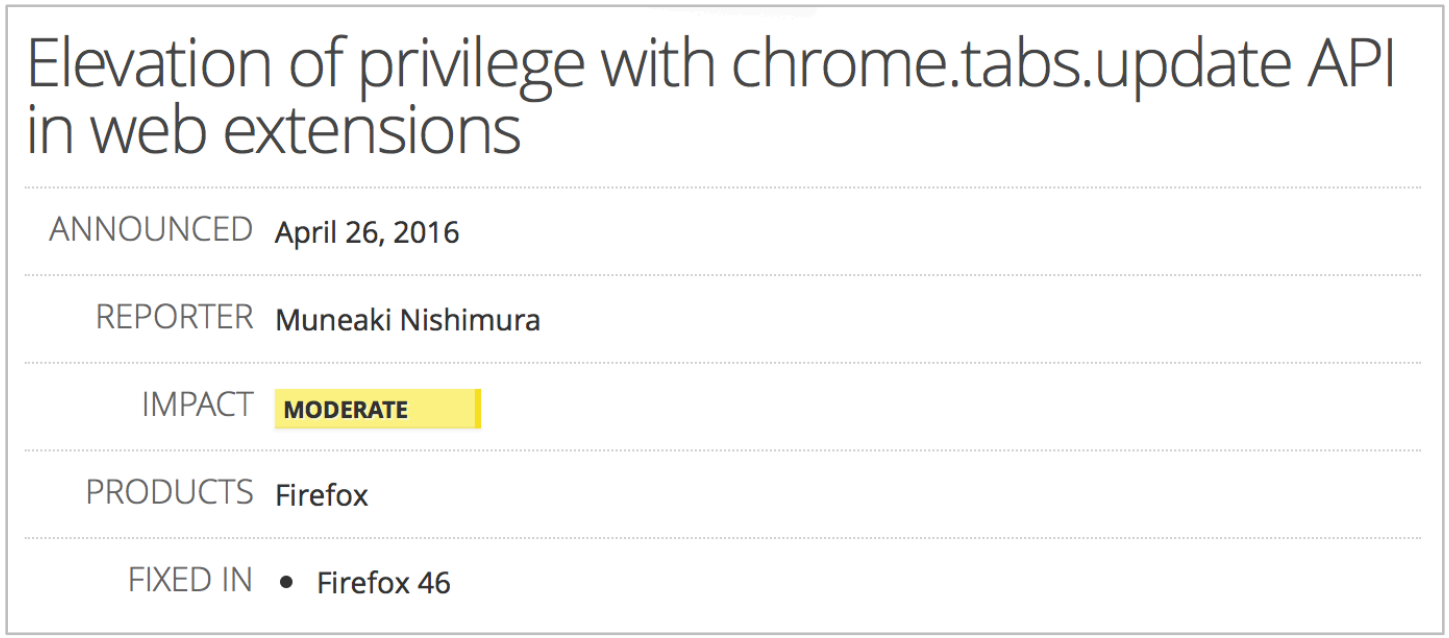
Contents

- **Simple one-time requests**
- **Long-lived connections** +
- **Cross-extension messaging**
- **Sending messages from web pages**
- **Native messaging**
- **Security considerations**
- **Examples**

https://developer.chrome.com/extensions/messaging

仕様に学ぶ - Web Extensionsの実装不備の事例

- FirefoxでChrome拡張機能を動作させる仕組み（Web Extensions）で**他の拡張に任意のJavaScriptを処理させる方法**を発見（CVE-2016-2817）
  - 従来のFirefoxアドオンでは許可された振る舞いだが、Web Extensionsでは脆弱性となる



Elevation of privilege with chrome.tabs.update API in web extensions

ANNOUNCED	April 26, 2016
REPORTER	Muneaki Nishimura
IMPACT	MODERATE
PRODUCTS	Firefox
FIXED IN	• Firefox 46

https://www.mozilla.org/en-US/security/advisories/mfsa2016-46/

過去に見つけた仕様の実装不備（抜粋）

- ChromeのCSP違反レポートの送信先が<base>で制御できる
  https://crbug.com/431218
- ChromeのService Workersが<iframe　sandbox>内で動作する
  https://crbug.com/486308
- FirefoxのReferrer Policyが新しいタブで開いた際に効かない
  https://lists.w3.org/Archives/Public/public-webappsec/2015Apr/0246.html
- FirefoxのBroadcast Channel API がプライバシーモードから
  通常モードのウィンドウに通知される
  https://bugzilla.mozilla.org/show_bug.cgi?id=1148032
- FirefoxのFetch APIでHostやCookieリクエストヘッダが指定できる
  https://bugzilla.mozilla.org/show_bug.cgi?id=1162411

発見者の目から見た脆弱性対応の現場

ブラウザベンダーの脆弱性対応

- 脆弱性の対応方針は**ベンダーによって**大きく異なる
- リモートコード実行（RCE）の脆弱性は重視して修正される一方、サンドボックスバイパス（同一生成元ポリシーの迂回など）の対応は温度感がベンダーによって異なる
  - ChromeとFirefox：脆弱性報奨金制度の対象。通常は1〜3ヶ月以内に修正
  - Safari：報奨金制度の対象外。1年以上修正されないことも

Firefoxの脆弱性対応

- 透明性が高い
  - 修正がリリースされるまでの過程を非公開設定のBugzillaで**追跡できる**
  - 深刻度の**判断理由を説明**してくれる
  - 他の誰かと同じ脆弱性（Duplicated）を報告すると、そう判断した証拠として、非公開設定になっている同じ脆弱性のBugzillaへのアクセス権をもらえる
- 対応が早い
  - 深刻度の高いものは約1〜3か月で修正
  - 深刻な脆弱性は**緊急アップデート**で修正

Firefoxの脆弱性対応 - 証明書検証バイパスの脆弱性事例

- Firefox 37で搭載された日和見暗号を用いて、HTTPSのサーバ証明書検証と公開鍵ピンニングを迂回できる脆弱性（CVE-2015-0799）を報告
- **3日後**、Firefox 37.0.1が緊急リリース

https://www.us-cert.gov/ncas/current-activity/2015/04/06/Mozilla-Releases-Security-Update-Firefox

日本の脆弱性窓口

- IPAの脆弱性関連情報の届出受付制度
  - 届出から最初の応答までに**約1ヶ月**
  - 担当者に指摘内容を理解して貰うまでに何度かやり取りが続く
  - 結果として、右図「2」までの間が長期化

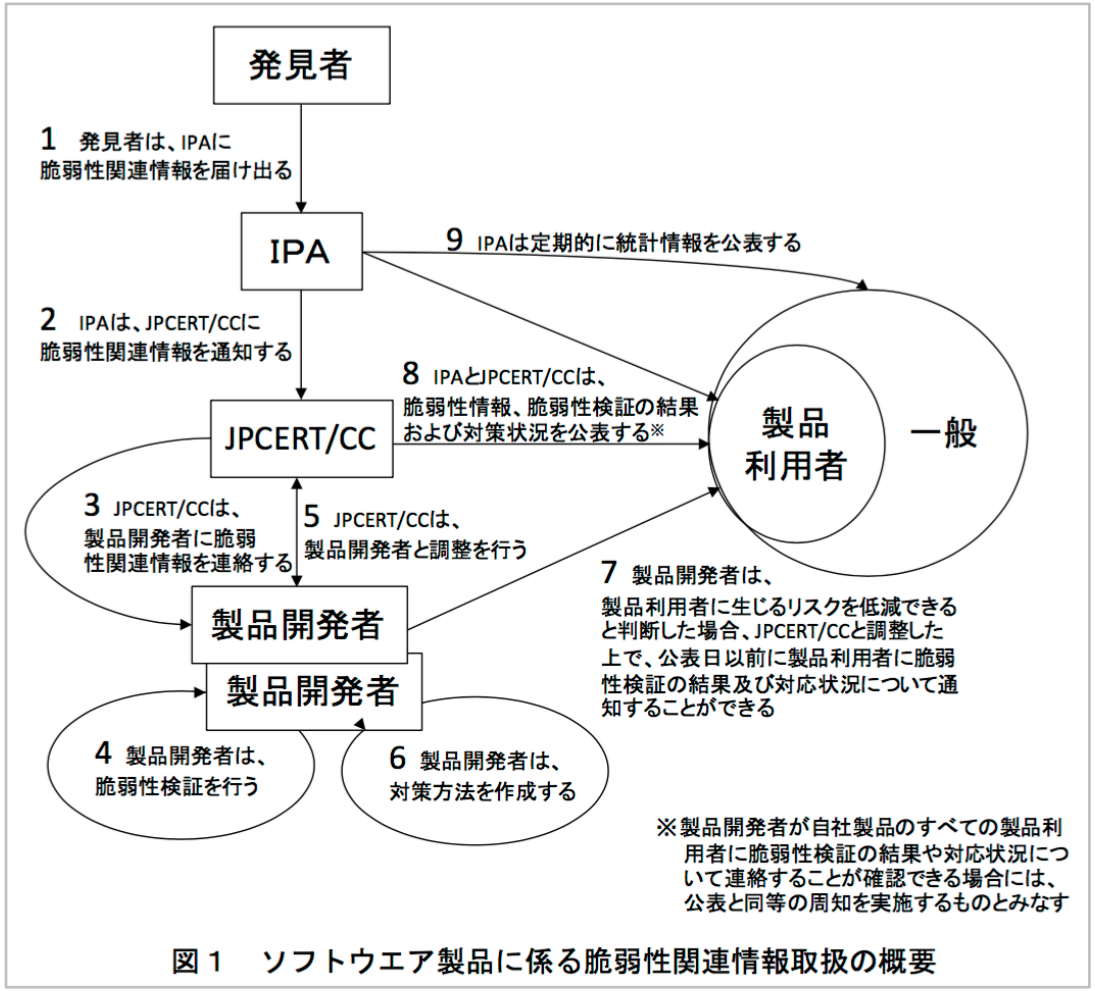


図１　ソフトウエア製品に係る脆弱性関連情報取扱の概要

https://www.ipa.go.jp/files/000052737.pdf

IPAの脆弱性窓口 – Adobe Cordovaの脆弱性事例

- Apache Cordovaの任意プラグイン実行の脆弱性（ CVE-2015-5208 ）
  - 悪意のあるサイトを開くだけで、アプリの機能を悪用される脆弱性。
    報告の際、**デモとして**スマホの電話帳を改ざんするコードを送付
  - 1ヶ月後、IPAから**電話帳が改ざんされることのどこが脆弱性なのか？**という質問
  - この時点で、JPCERT/CCに連絡が行われていない

https://jvn.jp/jp/JVN41772178/

IPAの脆弱性窓口 – Androidの脆弱性事例

- AndroidのHTTPヘッダインジェクション脆弱性（ CVE-2016-1155 ）
  - Androidの機能に脆弱性があることをIPAに報告
  - IPAでは脆弱性として扱わないので開発元に連絡して欲しいとの返答（**理由は不明**）
  - JPCERT/CCに直接連絡したところ、脆弱性として受理
  - JPCERT/CCの追加調査で、広範のAndroidに影響することが判明し**JVN公開**


公開日：2016/02/19　最終更新日：2016/02/19

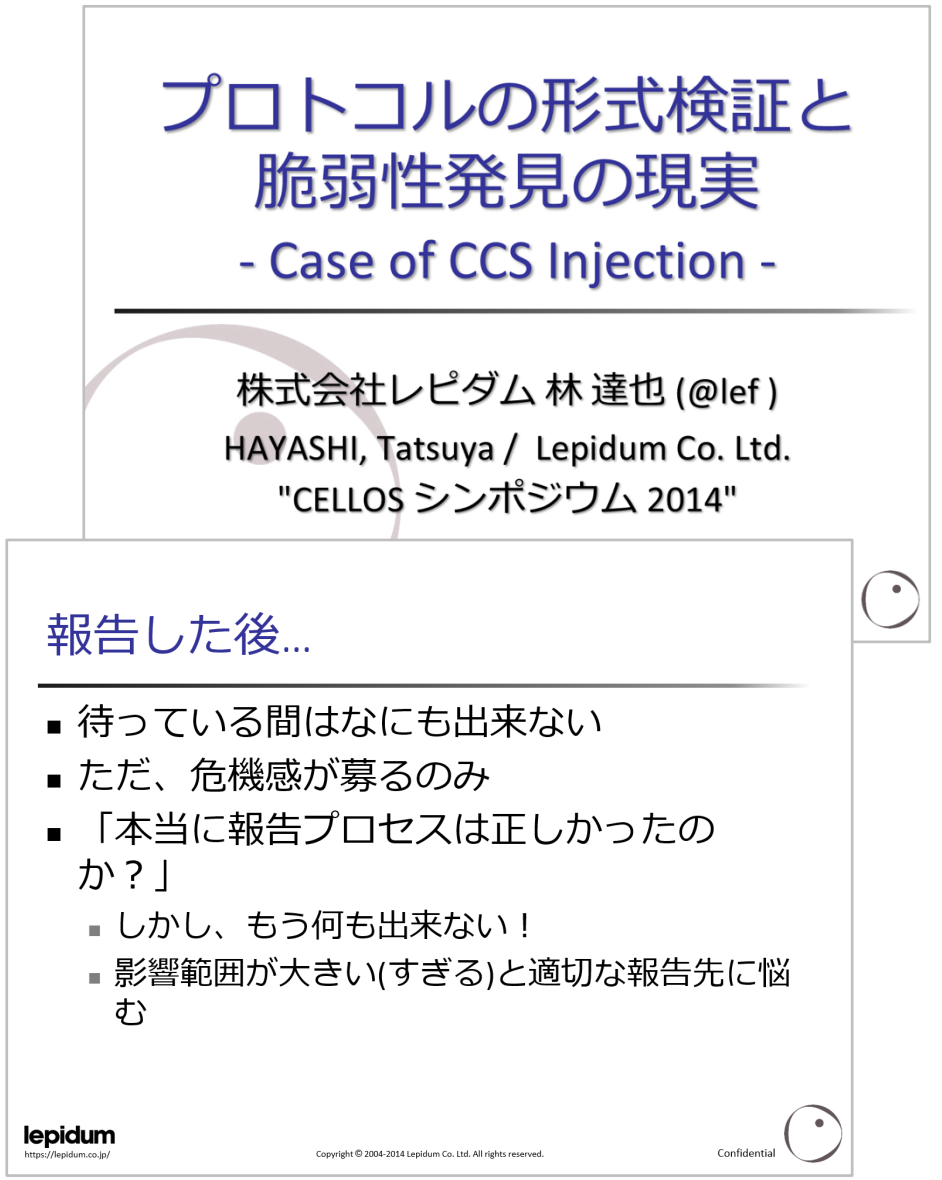
JVNVU#99757346
Android Platform の URLConnection クラスに HTTP ヘッダインジェクションの脆弱性

概要
Android Platform の URLConnection クラスには、HTTP ヘッダインジェクションの脆弱性が存在します。

https://jvn.jp/vu/JVNVU99757346/

発見者からのお願い

- 対応の**状況**を定期的に**共有して欲しい**
  - 対応完了まで一切連絡の来ない窓口が多い
  - 1年以上音信不通の事例もある
- 状況が分からないと**発見者は不安**になる
  - 報告の仕方が誤っていたのではないか（右図）
  - 早くしないと脆弱性が悪用されるのではないか
- 修正前の脆弱性が**公表される恐れ**も
  - 報告を無視されたと感じた発見者が怒って暴露
  - 公表により修正を促そうとする発見者も



※株式会社レピダム 林氏の講演資料より引用

リクルートにおける脆弱性対応

ガートナー セキュリティ & リスク・マネジメントサミット 2016 における鴨志田の講演資料より引用

Recruitについて –企業概要–

まだ、ここにない、出会い。 RECRUIT

世界中の生活者と産業界に「まだ、ここにない、出会い。」を提供。

創業	1960年
本社所在地	東京都千代田区
グループ従業員数	38,451名※
グループ企業数	287社※
連結売上高	約15,886億円※
連結経常利益	約1,193億円※

※2016年3月末時点

Recruitについて –沿革–

まだ、ここにない、出会い。
RECRUIT

1960年　広告代理店**大学新聞広告社**として創業

1963年　**株式会社日本リクルートセンター**へ社名変更

1984年　**株式会社リクルート**へ社名変更

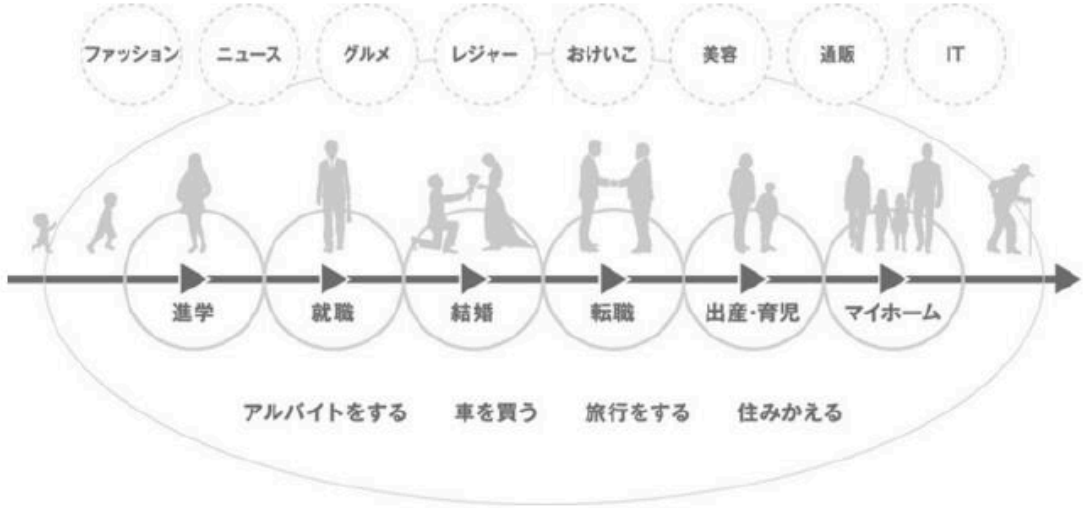
2012年　7事業会社、3機能会社からなる **リクルートグループ**へ体制変更

2014年　東証1部に**上場**

Recruitについて ービジネスモデルー

まだ、ここにない、出会い。 RECRUIT

選択・意思決定 を支援する情報サービスの提供。



リクナビ進学 / リクナビ / ゼクシィ / リクナビNEXT / 赤すぐ / suumo
HOT PEPPER グルメ / ケイコとマナブ / カーセンサー / じゃらん / AB-ROAD / HOT PEPPER Beauty
ERUCA / ATND / ポンパレモール / キーマンズネット / R25 L25 / Ciao

Recruitについて -ビジネスモデル-

まだ、ここにない、出会い。 RECRUIT

世界中の生活者と産業界に「まだ、ここにない、出会い。」を提供。

カスタマー
(一般ユーザー)

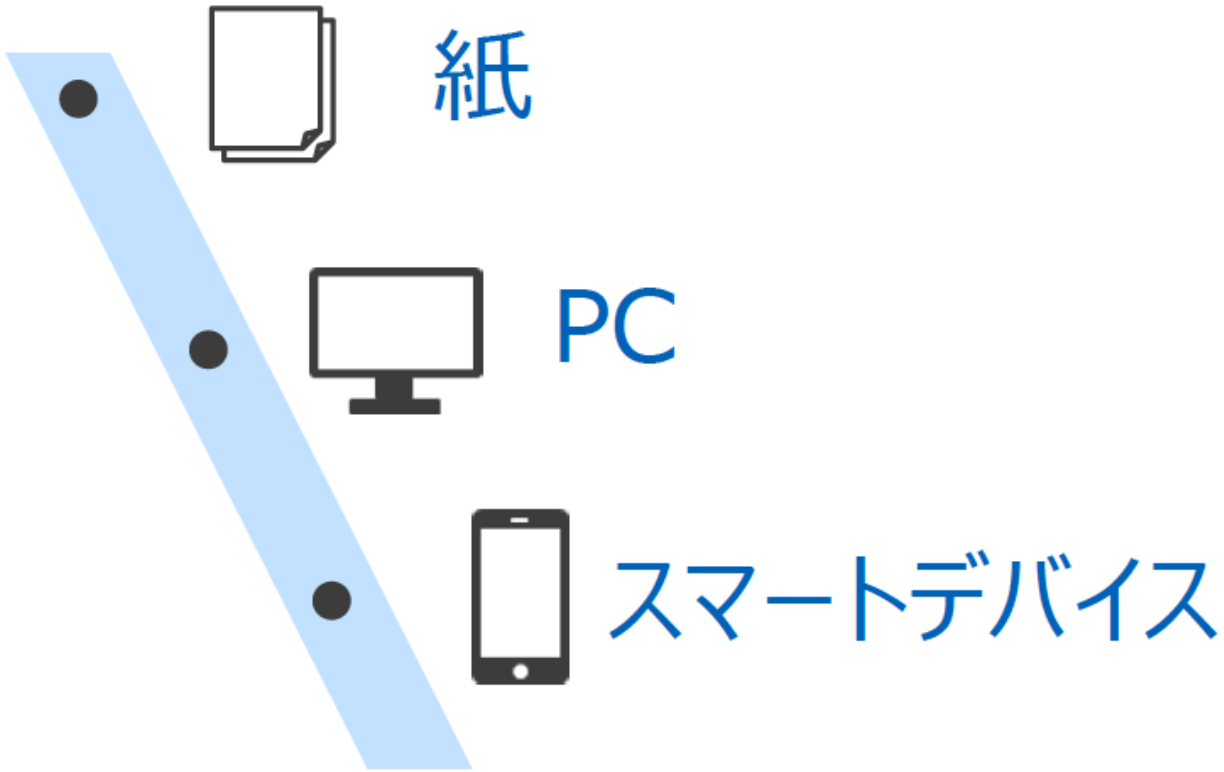
×
マッチング

クライアント
(サービス提供企業)

Recruitについて –ソリューションの変化–

RECRUIT まだ、ここにない、出会い。

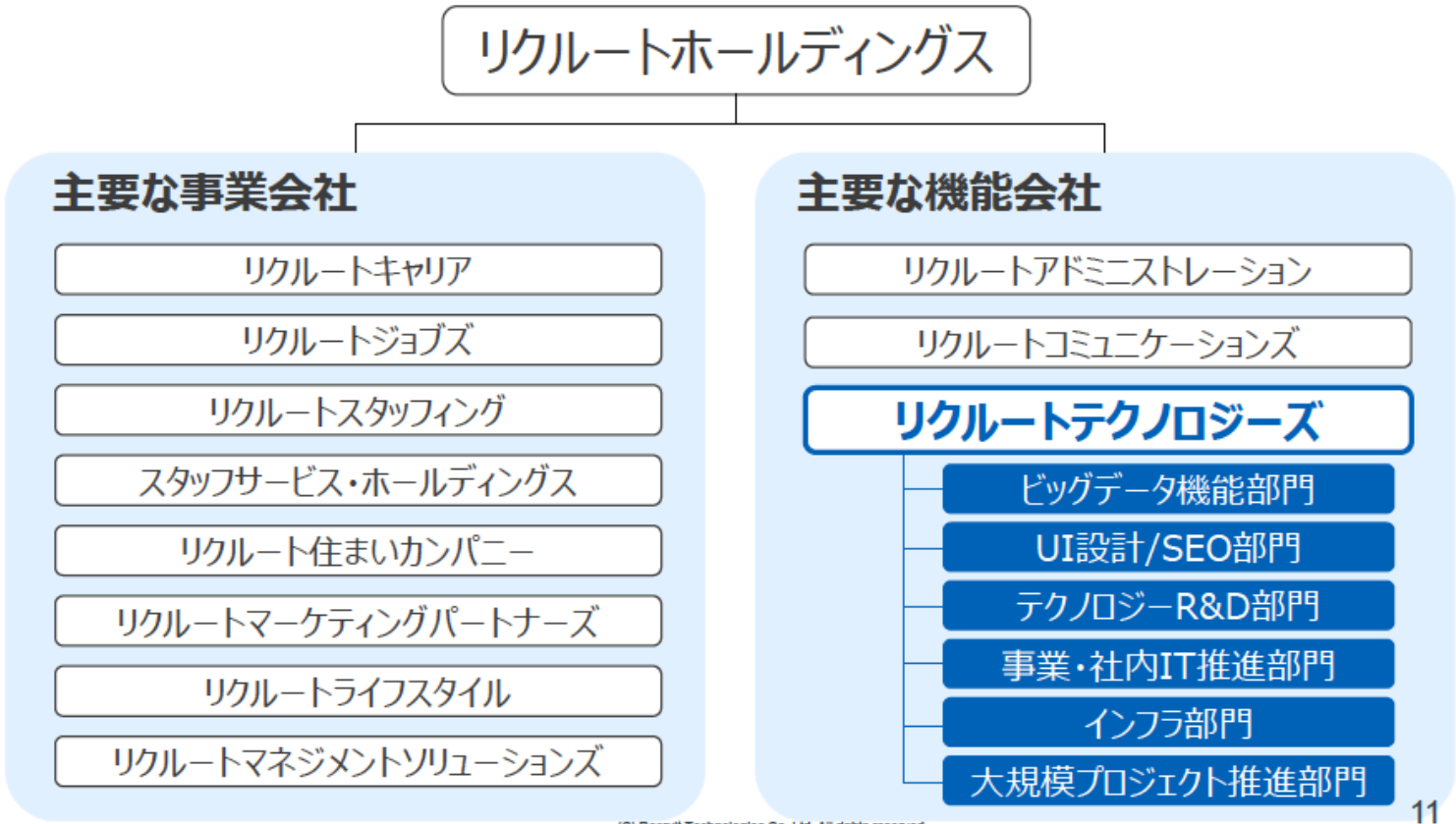
紙からネット への展開が促進。
ITの進化とともにソリューションの進化も求められる。



Recruitについて –構成–

RECRUIT まだ、ここにない、出会い。

リクルートグループは複数の事業会社、機能会社から構成。



Recruitについて –Recruit Technologiesについて–

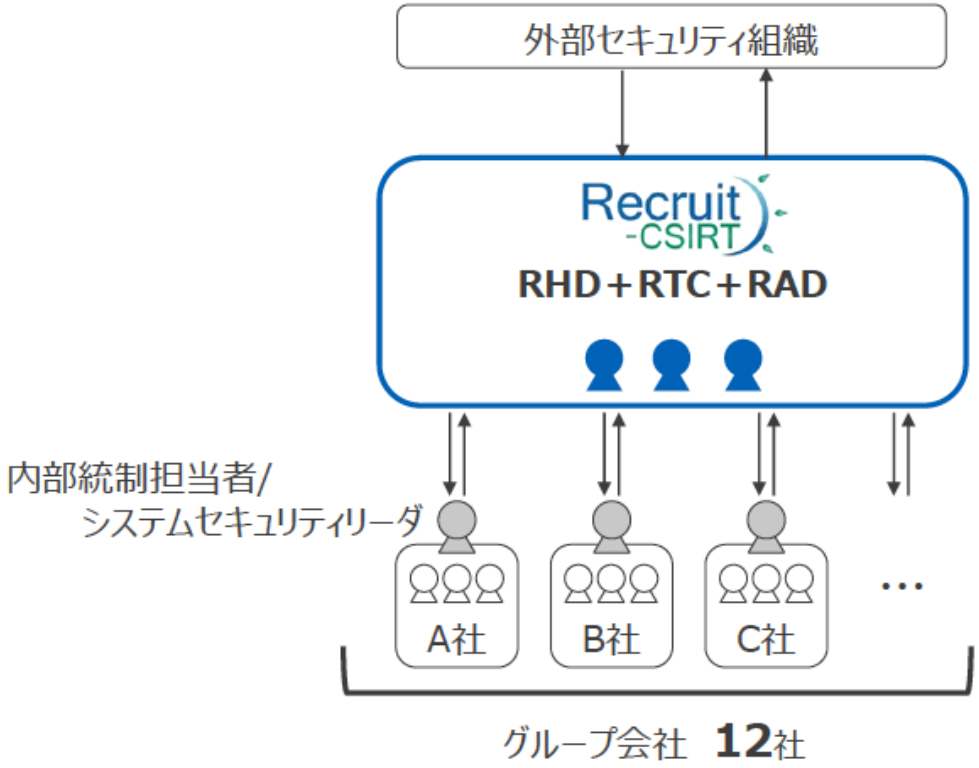
まだ、ここにない、出会い。 RECRUIT

リクルートテクノロジーズは、IT・ネットマーケティング領域の専門部隊であり、
リクルートグループをITで牽引する企業。

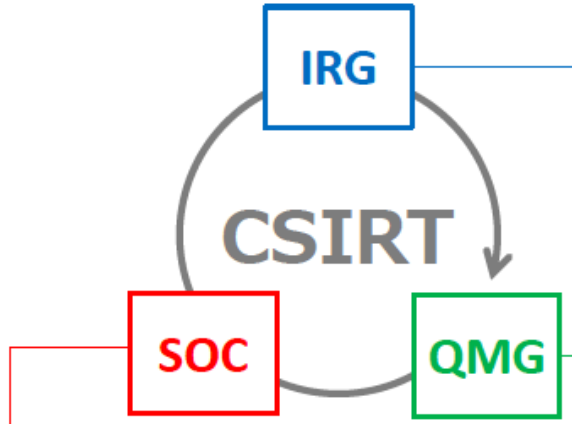
2. Recruit-CSIRTのご紹介 –構成-

まだ、ここにない、出会い。 RECRUIT

セキュリティ専任部隊 Recruit-CSIRTを設立し、2015年度より本格始動。
リクルートホールディングス(RHD)、リクルートテクノロジーズ(RTC)、リクルートアドミニストレーション(RAD)により構成。



2. Recruit-CSIRTのご紹介 –RTC内組織-



IRG
Incident Response Group

被害最小化

サイバー攻撃への対応をリード

- 事故発生時の対応支援
- 外部関連機関との連携
- 早期警戒(脅威情報の収集および対策検討)
- Recruit-CSIRTの運営、各社展開

SOC
Security Operation Center

早期検知

サイバー攻撃を早期検知し、被害拡大を防止

- グループ共通インフラのセキュリティ監視
- 未知マルウェアの監視、解析、一次対応
- 被害発生時のフォレンジック (ネットワークおよびPC)
- 内部不正のモニタリング

QMG
Quality Management Group

未然防止

平時からセキュリティを向上、被害を未然に防止

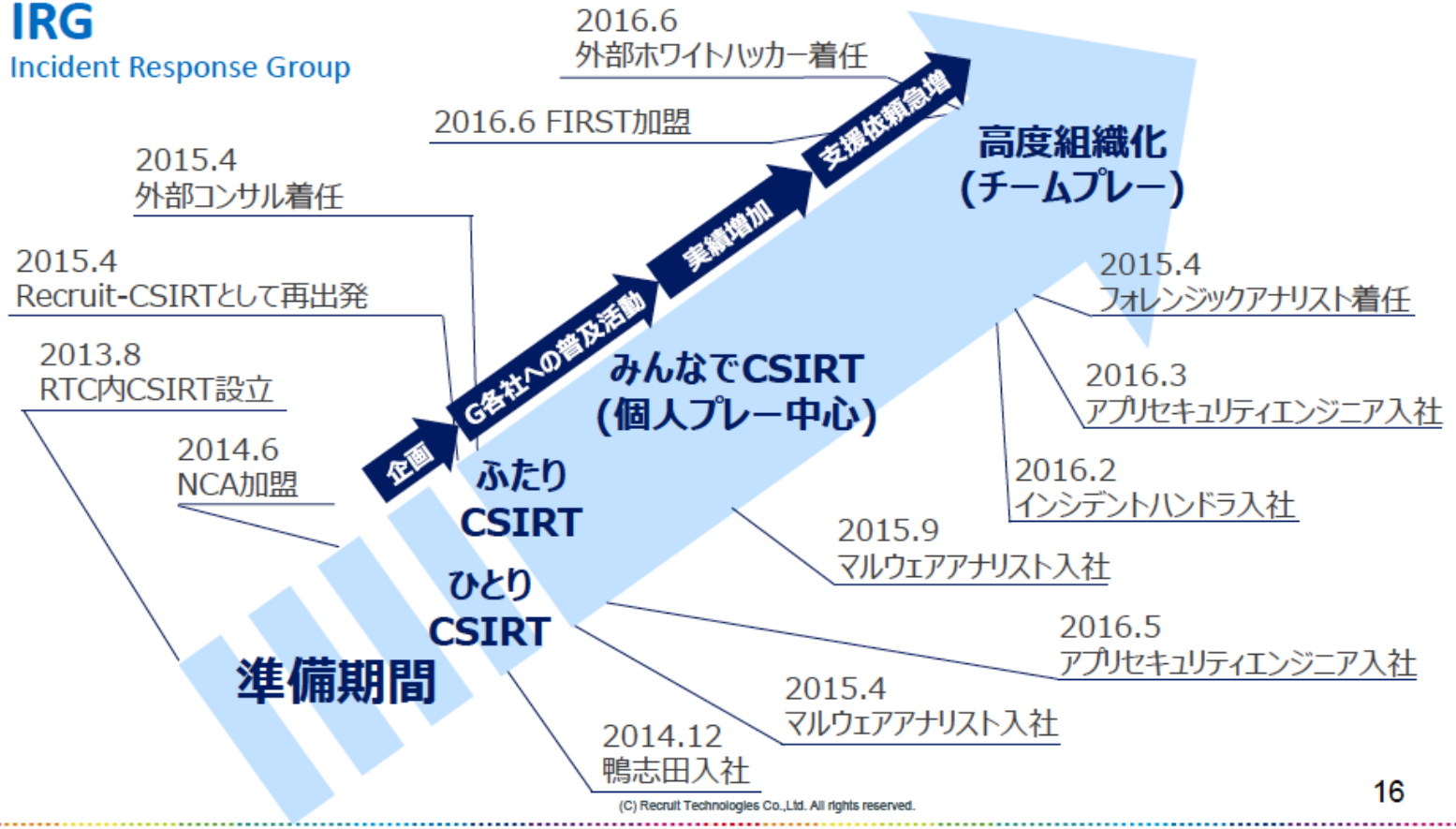
- 脆弱性診断、開発者教育などのセキュア開発支援
- セキュリティパッチの情報収集および各社展開
- システム運用の簡易監査

2. Recruit-CSIRTのご紹介 –IRG沿革-

RECRUIT まだ、ここにない、出会い。

4か月の企画を経て、優秀なメンバーを集めながら1年で高度な技術を持つグループCSIRTを構築

IRG
Incident Response Group



2. Recruit-CSIRTのご紹介 –IRGメンバ-

まだ、ここにない、出会い。 RECRUIT

IRG
Incident Response Group

1

組織マネージャ

IRG工数
★★⯪☆☆

2

インシデント ハンドラ

IRG工数
★★★★★

3

マルウェア アナリスト

IRG工数
★★☆☆☆

4

マルウェア アナリスト

IRG工数
★☆☆☆☆

5

フォレンジック アナリスト

IRG工数
★☆☆☆☆

6

アプリセキュリティ エンジニア

IRG工数
★★⯪☆☆

7

外部 コンサルタント

IRG工数
★★⯪☆☆

8

外部 ホワイトハッカー

IRG工数
★★☆☆☆

2. Recruit-CSIRTの人材方針 –人材の内製化-

まだ、ここにない、出会い。 RECRUIT

サイバー攻撃の増加に耐えられるよう、セキュリティ人材を確保/教育し、
CSIRT活動の核を内製にて行うことを目指している。

世間

サイバー攻撃 **増** × セキュリティ人材 **少**

方針

CSIRT活動の核について、
内製にて実施できるよう、セキュリティ人材をそろえる。

リクルートにおける脆弱性対応の基本方針

- 脆弱性対応は各事業で実施
  - **緊急性の高い脆弱性のみ**、CSIRTから各事業へ対応を依頼
- 脆弱性情報の収集と深刻度の評価をCSIRTで実施
  - 私を含め3人の技術者が持ち回りで担当

一般的な脆弱性情報提供サービスの課題

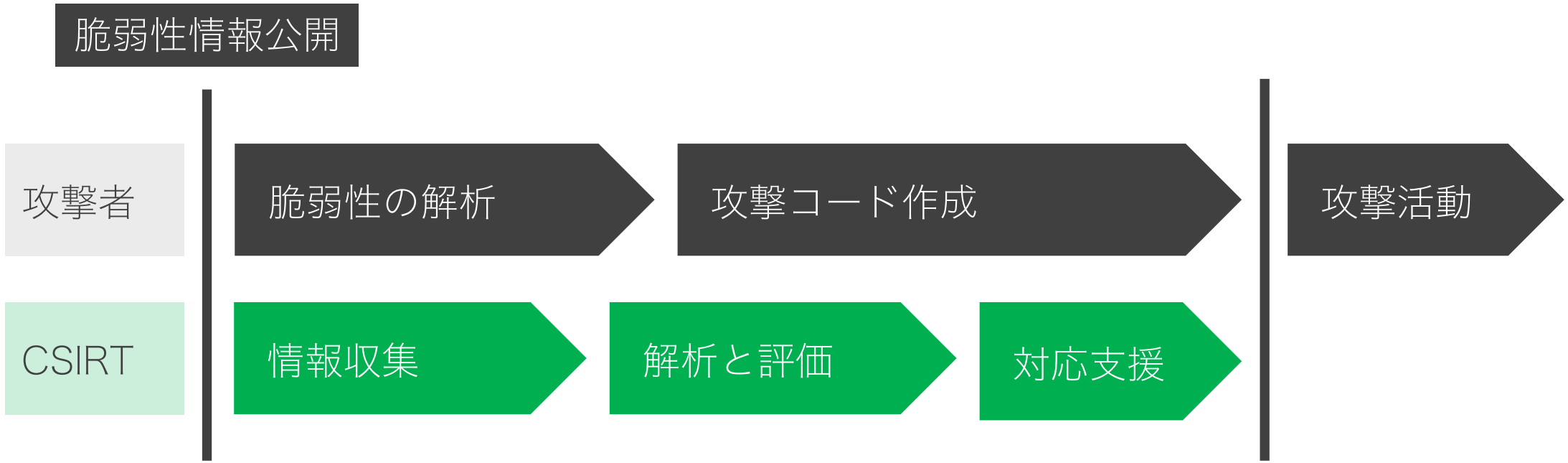
- 情報配信の**タイムラグ**
  - 開発元による情報公開から半日程度の遅延
- 情報の**網羅性**
  - CVEの割り当てられていない脆弱性などに抜け
- 自社環境と一致しない**深刻度評価**
  - CVSSは脆弱性が最大限に悪用された前提で見積もられる傾向があり、自社における深刻度とは一致しないことがある

そこで、自社で脆弱性情報の収集と評価が必要

- 情報展開の**早期化**
  - 脆弱性の一次情報を徹底収集
  - JPCERT/CCの早期警戒パートナーシップを通じて、公表前の脆弱性情報を入手
- **網羅性**の向上
  - 国内外のセキュリティ情報発信者をTwitterでフォロー
  - JPCERT/CCの早期警戒情報を用いて、情報の収集漏れを低減
- 自社環境に基づいた**脆弱性評価**
  - 脆弱性をCSIRTで解析し、攻撃の難易度や自社で生じうる被害を評価

情報展開の早期化

- 攻撃が来る前に、情報の収集、評価、対応支援を終わらせる必要がある
  - 2014年のShellshockのように情報が公開された翌日から攻撃が観測された事例も



情報展開の早期化

- **4つの情報経路**を組み合わせて、情報収集の早期化と網羅性を追求
  - 二次情報で情報を得たものは、一次情報で収集できるようにフィードバック

脆弱性情報公開

JPCERT/CC 早期警戒パートナーシップ

- 公開前の脆弱性情報

一次情報

- 各種RSSやML
- 開発元のWebサイト

二次情報

- Twitter
- 各種ニュースサイト

JPCERT/CC 早期警戒情報

- 毎日夕方に来る アナリストノート

情報展開の早期化 – Ghostscriptの脆弱性事例

- Ghostscriptにおけるリモートコード実行の脆弱性（CVE-2016-7976）

日時	状況
2016年9月30日	Ghostscriptの開発元に脆弱性の投稿（**実証コードあり**）
2016年10月5日	**OSS Security MLでCVE採番**
2016年10月13日	国内の脆弱性情報サービスに情報掲載
2016年10月14日	MetasploitのGitHubにて本脆弱性の攻撃コードを確認
2016年10月14日	JPCERT/CCより早期警戒パートナーシップへ情報配信
2016年10月17日	JPCERT/CCより早期警戒情報配信

この時点で各事業に緊急対応を依頼

情報展開の早期化 – その他の事例

- Apache Struts2
  - 公開前のアドバイザリが公開設定のConfluenceに何故か置いてある
    http://struts.apache.org/docs/security-bulletins.html
  - Confluenceからの情報入手が最速
- OpenSSL
  - ML（openssl-announce）の情報展開が最速
  - 公式ブログではほとんど情報配信されないので注意
  - BindやVMWareなど、MLでのアナウンスが優先される製品は多い

自社環境に基づいた脆弱性評価

- 公開されている**CVSS値**を**鵜呑みにしない**
  - 影響しない脆弱性の対応に各事業の工数を割くことになる
  - 自社環境における影響を技術的に判断
- 可能な限り修正パッチの**ソースコードを確認**
  - 攻撃の容易性を推測できる
  - 追加されたテストケースやコミットメッセージに攻撃コードが含まれることも
- **攻撃ツール**の開発**リポジトリを監視**して攻撃の可能性を判断
  - MetasploitのGitHubに攻撃コードが続々と集まる
    https://github.com/rapid7/metasploit-framework/pulls

自社環境に基づいた脆弱性評価

- セキュリティ技術者が集まる**Slackコミュニティ**の活用
  - 所属組織の壁を超えて脆弱性の解析状況を共有
  - セカンドオピニオン的に活用し、解析の誤りを防ぐ

自社環境に基づいた脆弱性評価 – CVE-2016-0799

- OpenSSLにおけるメモリ操作の不備の脆弱性
- NIST評価のCVSSベース値は10.0
  - メモリ破壊系の脆弱性の被害は通常サービス妨害。
    メモリの壊し方次第でリモートコード実行となる
  - 攻撃の複雑さ（低）はサービス妨害を前提に
    見積もられている一方、影響（全面的）は
    リモートコード実行を前提としている
- OpenSSLのアドバイザリ（一次情報）では
  - Webサーバやロードバランサが**HTTPS通信時に**
    使用する**libsslには影響しない**

一次情報より緊急対応不要と判断

NISTのCVSSv2ベーススコア

メトリクス	値
攻撃ベクタ	ネットワーク
攻撃の複雑さ	低＝簡単
認証の要否	認証なし
機密性への影響	全面的
完全性への影響	全面的
可用性への影響	全面的

自社環境に基づいた脆弱性評価 – CVE-2016-2776

- Bind9におけるサービス妨害の脆弱性

日時	状況
2016年9月27日	開発元から脆弱性情報公開。開発リポジトリに修正パッチあり。 解析の結果、DNSリクエスト1発でDNSサーバを 落とせることが判明。**緊急対応を実施**
2016年10月1日	MetasploitのGitHubに攻撃コードの存在を確認
2016年10月3日	**主要サービスにおいてパッチ適用完了**
2016年10月5日	警察庁より、本脆弱性を悪用した無差別な攻撃活動が 監視されたとの注意喚起

予定している取り組み

- 情報収集の**効率化**
  - Twitterによる情報収集（毎日30分〜1時間）の置き換えを検討中
- **攻撃活動**の情報収集を強化
  - 各国の攻撃発生状況を収集
  - 攻撃の発生状況に応じて、速やかに対応の温度感を変える
- **各サービスの構成**に応じた対応依頼
  - サーバ構成やコンフィグレーションなどを詳細に把握し、
    外部から攻撃を受ける可能性に応じて対応の温度感を分ける

まとめ

脆弱性を探し始めた理由と続ける理由

- 獲得した報奨金の額で自分の**実力を可視化**
- 前線で通用する**技術力の維持向上**

脆弱性を見つけるために実践していること

- **過去の脆弱性や仕様**に学ぶ
- 日頃から脆弱性の情報を収集し、**検証する習慣**を付ける

発見者の目から見た脆弱性対応の現場

- 対応の仕方はベンダーによって様々
- 脆弱性対応の現場は、**発見者の気持ちを考える**ことが大切
  - 報告する／される側の双方を経験してみると良い

リクルートにおける脆弱性対応

- 情報収集の**早期化と網羅性**を追求
  - 4種類の情報経路を組み合わせる
- **自社環境における深刻度**を評価
  - 一次情報だけでなく、修正パッチの中身も解析
  - 攻撃ツールのリポジトリをモニタリングし、攻撃コードの出現を監視
  - セキュリティ技術者が集まるSlackを通じて、脆弱性情報をより深く理解

さいごに

- **業務と個人の活動のシナジー**を強化させていきたい
  - 業務で脆弱性を解析することで、脆弱性を見つける力がつく
  - 個人の活動で脆弱性を見つける力を身につけて、自社のセキュリティ品質向上に貢献

おわり