セキュリティ品質を考慮したSDLにおけるQAの役割(テスト・評価編)

１．背景
２．前期（８期）のふりかえり
３．テスト・評価フェーズのセキュリティ品質保証
４．実践編
５．まとめ

第9期ソフトウェア品質保証部長の会　第6グループ

株式会社モバイルインターネットテクノロジー　二川　勇樹
キヤノンＩＴソリューションズ株式会社　日下　宏
キヤノンOB/日科技連ソフトウェア品質保証部長の会企画委員　永田　哲
(順不同)

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

1. 背景（1）

■セキュリティ品質の要求の高まり

・あらゆるシステムがネットワークに繋がり、システム同士の連携も増え、
セキュリティ上の脅威が格段に増大。
繋がった先のシステムの脆弱性等、考慮すべき範囲も拡大。

・国家レベルでのサイバーテロ等、ハッカーの目的も変わってきており、
システムのセキュリティ品質の確保の必要性が益々高まっている。

・世の中の変化の加速で、考えるべきセキュリティリスクも変化。
最もクリティカルと考えられるセキュリティリスクをまとめた
「OWASP Top10」にも、「XML 外部エンティティ参照」や
「安全でないデシリアライゼーション」等、新しい脆弱性が取り上げられ
ている。

**セキュリティ品質を担保する為に、経営者から企画、開発、
品質保証まで、全社的なセキュリティ対策が必要な時代に！**

Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

1. 背景（2） OWASP TOP10 2017

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – インジェクション	→	A1:2017-インジェクション
A2 – 認証の不備とセッション管理	→	A2:2017-認証の不備
A3 – クロスサイトスクリプティング (XSS)	↘	A3:2017-機微な情報の露出
A4 – 安全でないオブジェクトへの直接参照 [A7とマージ]	U	A4:2017-XML 外部エンティティ参照 (XXE)[NEW]
A5 – 不適切なセキュリティ設定	↘	A5:2017-アクセス制御の不備 [マージ]
A6 – 機微な情報の露出	↗	A6:2017-不適切なセキュリティ設定
A7 – 機能レベルのアクセス制御の不足 [A4とマージ]	U	A7:2017-クロスサイトスクリプティング (XSS)
A8 – クロスサイトリクエストフォージェリ (CSRF)	☒	A8:2017-安全でないデシリアライゼーション [NEW,コミュニティ]
A9 – 既知の脆弱性のあるコンポーネントの使用	→	A9:2017-既知の脆弱性のあるコンポーネントの使用
A10 – 未検証のリダイレクトと転送	☒	A10:2017-不十分なロギングとモニタリング[NEW,コミュニティ]

出典:「OWASP Top10 2017 最も重大なウェブアプリケーションリスクトップ10」

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

1. 背景（3）

■セキュリティ品質を担保する為のQAの役割

・セキュリティを担保するためにはQAは、設計や実装の良し悪しを
判断できなければいけない。
その為にはセキュリティに関する「リテラシー」や「スキル」が必須。

・脆弱性診断のエキスパートに見てもらうことは重要だが、時間的にも
コスト的にも限界有。
その前段階で、自前でセキュリティレベルを上げておくことで、
セキュリティ品質の底上げにつなげられる。

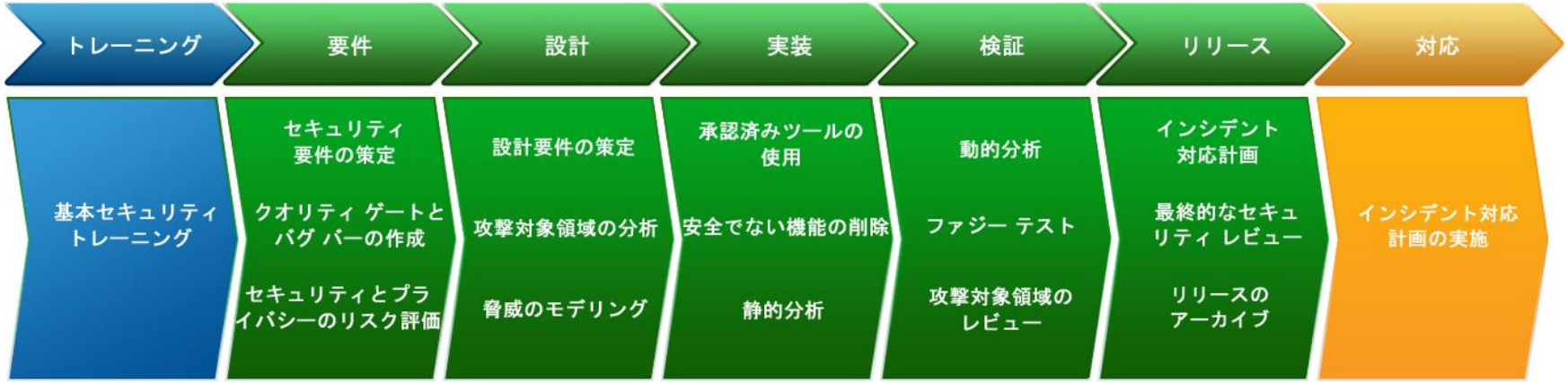
SQiP
Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

2. 前期(8期)のふりかえり（1）

■「セキュリティ開発ライフサイクル（SDL）における QAの役割」

2017年の品質シンポジウムの日下氏の発表にて、
WEBアプリケーションの開発においてソフトウェア品質保証や
品質評価部門（QA）がセキュリティ品質をどう担保すべきか、
セキュリティデベロップメントライフサイクル（SDL）をベースに
我々の考えをお伝えしました。



出典:「マイクロソフト社Microsoft SDL の簡単な実装
Japanese Simplified Implementation of the SDL.docx 」

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

2. 前期(8期)のふりかえり（2）

■「Thread Modeling ToolsやCVSS v 3の提案」

- セキュリティ品質の確保には、上流工程で対策が非常に重要
- 後工程だと、コストもかかり、工期的に十分な改修がされないことも。
- 8期では上流工程にフォーカスし、脅威分析と、リスクマネジメントの重要性をお話し、品質保証部門の下記取組みを提案。

1. DFD図を用い、「守るべき情報・機能資産」を明確化
2. Microsoft社のThread Modeling Toolsを脅威分析
3. リスクマネジメントチェックシートとCVSSv3を用いて
   リスクの評価とリスクの低減を実施

上流工程からの脅威分析／リスクマネージメントが全体のセキュリティ品質の骨格を形成

SQiP Software Quality Profession

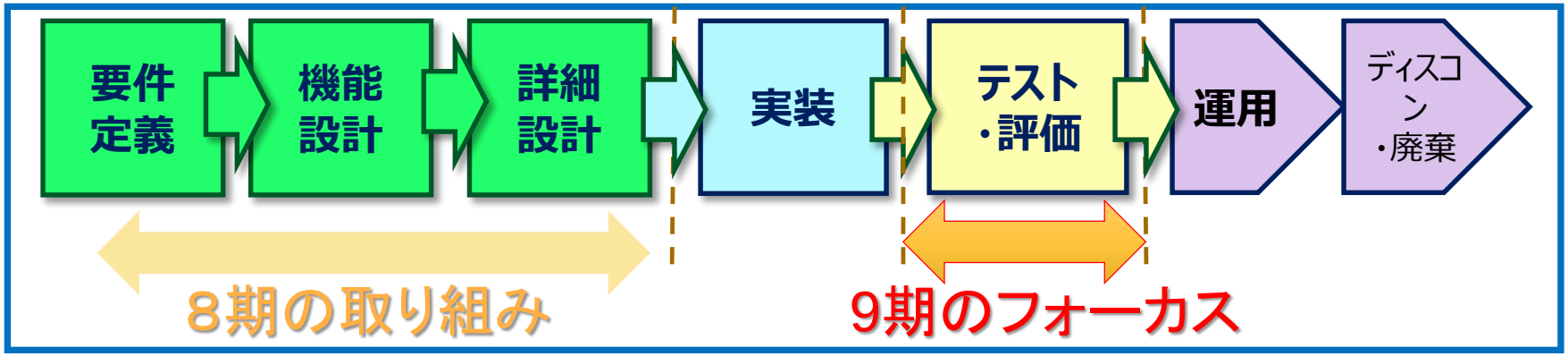
Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

3．テスト・評価フェーズのセキュリティ品質保証（1）

■9期セキュリティグループの活動

・9期はソフトウェア開発ライフサイクルのうち、テスト・評価フェーズにフォーカスし、下記観点でセキュリティ品質確保方法を検討し、実戦で利用します。

- 「品質保証部門によるセキュリティ品質保証の考え方」
- 「脆弱性テスト及び脆弱性評価の方法と実例」



※「実装」フェーズは、開発部門が中心に実施し、「テスト・評価」フェーズで、品質保証部門がその実施状況を確認することを想定し、フォーカス外に。

SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

3．テスト・評価フェーズのセキュリティ品質保証（2）

■品質保証部門のセキュリティ品質保証の考え方

QAの活動項目	活動の目的	QAに必要な能力
[上流工程] セキュリティ品質の作りこみの確認	・設計からセキュリティを作り込めているかを確認 →後工程での手戻りや脆弱性が残る可能性を低減	・QAは開発者のセキュリティ設計や脆弱性対策の妥当性を評価できる能力 →ツールを用いた脅威分析とリスクマネージメント
[テスト・評価フェーズ] 実装結果の妥当性評価 →実装時の脆弱性対策が万全か？	・開発～運用まで単純な脆弱性のテストは、自分たちで継続的に実施 ・脆弱性診断のエキスパートには、難しい脆弱性の調査に専念してもらい、ROIを挙げる	・自前で脆弱性テスト/診断が行い、妥当性を評価できる能力 →ツールを用いた脆弱性診断（手動/自動）

SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

3．テスト・評価フェーズのセキュリティ品質保証（3）

■テスト・評価フェーズにおけるQAの活動

・**評価基準を定める上で各種ガイドラインや書籍を学習し、実戦利用。**

「IPA　Webアプリケーション脆弱性対策チェックリスト」

品質保証部門向けにWebアプリケーション脆弱性チェックリスト

「IPA　安全なウェブサイトの作り方（第7版）」

それぞれの脆弱性で発生しうる脅威や特徴等を解説。

根本的な解決策/攻撃による影響の低減のための対策を記述。

「Webセキュリティ担当者のための脆弱性診断スタートガイド
上野宣が教える情報漏えいを防ぐ技術　(翔泳社)」

書籍中にWebアプリケーションの脆弱性チェックリストの紹介あり。

ガイドラインやチェックリストに沿って評価

SQiP
Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

3．テスト・評価フェーズのセキュリティ品質保証（4）

■脆弱性テストの流れ

1）テストケース作成

脆弱性チェックリストを利用し、テストケース作成

2）自動診断ツールによる診断

ポートスキャナによるオープンなポート探索
OS/ミドルウェアの脆弱性診断
Webアプリの脆弱性診断

3）Proxyツールによる手動診断

自動診断で出てきた問題の調査や評価
自動診断では見つけられない脆弱性を手動で確認

効率的な脆弱性評価（診断）の仕組みを構築するには、
目的に応じた手法を適材適所に活用することが有効。

SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

脆弱性チェックリストを用いて、テストケース作成

公開されている脆弱性チェックリストを元にテストケース作成。

※IPAの「 Webアプリケーション脆弱性チェックリスト」等

危険度	対策テーマ	場面	点検事項	セミナスライドの該当箇所	点検結果（○・×・外）	備考（明細、除外理由 等）
高	3.1 SQL注入対策	3.1.1 リスク対策に関わる属性	危険度=高; 要警戒箇所=多; 複雑さ=中; 対策を行う工程=主にプログラミング工程; 対策共通部品の効果=高い; セミナー区分=第3回。	3-1-1 SQL注入#1 実装における対策		
		3.1.2 予備質問	(1) 使用するDBソフトウェアの種類とバージョンは何か？			
			(2) DBのプログラミングにおいて、プログラマが直接SQL文を取り扱うか？ またはO/Rマッピング技術等を用いてDBにアクセスするか？			
			(3) DBソフトウェアへアクセスするためのAPIライブラリには何が用いられるか？			
			(4) そのAPIはプリペアドステートメントをサポートしているか？			
			(5) そのプリペアドステートメントAPIは、APIライブラリ内部で特殊記号のエスケープ処理を行う擬似的なものではないことが確認できているか？			
			(6) DBソフトウェアに日本語データを渡す際、どの文字コードが用いられるか？			
			(7) 日本語を表すマルチバイトデータの途中に 0x27「'」や 0x5C「¥」のビットパターンが含まれているとき、DBソフトウェアがこれらをASCII文字「'」や「¥」と誤って認識してしまうことがないか？			
		3.1.3 入力パラメータ値のSQL文への埋め込みの際の対策				
		3.1.3.1 DBのプログラミングにおいて、プログラマは直接SQL文を取り扱わず、O/Rマッピング技術等を用いてDBにアクセスする場合				
			(1) 当該O/Rマッピング等の技術の実装において、SQL注入対策が施されているか？ すなわち、下記1.3.2以降の対策が施されているか？			
		3.1.3.2 DBのプログラミングにおいて、プログラマが直接SQL文を取り扱う場合				
		3.1.3.2.1 DBソフトウェアのAPIが擬似ではないプリペアドステートメントをサポートしている場合	(1) 開発プロジェクトのコーディング規約に、SQL文へパラメータを埋め込む際、プリペアドステートメントを使う旨の規定があるか？			
			(2) 文字列連結によってSQL文の途中にパラメータが埋め込まれる箇所が一箇所も無いようソースコードが記述されているか？			
			(3) プリペアドステートメント機能を用いて埋め込むことのできない文法要素を可変にする目的で文字列連結演算が使用されている場合、埋め込まれる値が、信頼できないソース（Webクライアントからのパラメータ、外部から受信したファイルの内容の一部等）からのものでないことが確実になっているか？			

出典:「IPA Webアプリケーション脆弱性チェックリスト」

Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

自動診断ツールと、手動診断の特徴

	メリット	デメリット・注意点
自動診断ツール	・スキルが低くても広範囲の診断が可能 ・診断時間が短く、繰返し容易 ・人的負担が少ない	・システム構成やツール特性により誤検知、見落としあり ・システムで特定条件が必要な脆弱性は検知できない ・脆弱性情報の更新頻度の高いツールの選定が必要
手動診断	・柔軟性が高く、細やかな診断が可能 ・システムの様々な条件を変えながら診断が可能	・セキュリティ・NW・IT技術の専門性の高い要員が必要。 ・限られた「脆弱性診断のエキスパート」が時間をかけて診断する為、アサインが難しい。

コスト、時間、人員などの条件を総合的に検討し、適した手法を選択。
全体をツールでカバーしつつ、高リスクかつ重要な箇所のみ手動で診断する、
最初にツールで診断した結果をもとに手動で再確認する場所を絞り込むなど、
ツールを上手く活用すると、費用対効果の高い脆弱性診断になる

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

脆弱性テストの構成例

QAの立場から脆弱性対策を行う為に、実践的なトレーニングが必須。ここでは下記ツールを利用。

1. **ポートスキャナ**　NMAP
2. **脆弱性スキャナ**　OWASP ZAP　/ NESSUS(有償)
3. **Proxyツール**　Burp Suite Community Edittion
4. **攻撃対象サーバ** BadStore / EasyBuggy

今回のツール選定基準

「UIベース等、QAが取り扱いしやすい」
「高頻度で脆弱性情報が更新される」
「無償もしくは、有償でも無償でトライアルで確認可能」
※他にも下記のようなツールが有名。
　脆弱性スキャナ　AppScan(IBM)、WebInspect(HP)　※有償
　攻撃対象サーバ　metastable2

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

４．実践編（1）

■　脆弱性テストの実践

実践編として、以下のツールによる脆弱性テストの概要を紹介する。

１）ポートスキャンによるオープンなポート探索(Nmap)

２）OS/ミドルウェアの脆弱性の自動診断（Nessus）

３）Webアプリの脆性性の自動診断（OWASP ZAP)

４）Proxyツールを用いた手動診断（Burp Suite)

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．実践編（2）

1）ポートスキャンによるオープンなポート探索(Nmap)

- **ポートスキャナNmapによる脆弱性検査**
- **サーバーの基本的なセキュリティ対策**
  - マシンがどのポートでサービスを提供しているのかを把握するためにネットワークプロトコルのトランスポート層に対してポートスキャナツールで調査する。
- **ポートスキャン**
  - 対象として指定したホストに対してポート番号を変えながらIPパケットを送信し、その反応を調べ、どのポートが外部からアクセス可能か調査する。

※公のネットワーク上の他者が管理しているサーバに対しポートスキャンを行うことは不正アクセスに該当する可能性がある。Nmapを利用する際は十分に注意が必要。

Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．実践編（3）

1）ポートスキャンによるオープンなポート探索(Nmap)

コマンドプロンプトでポートスキャンを実施。

①指定したIPアドレスに対し、ポートスキャン

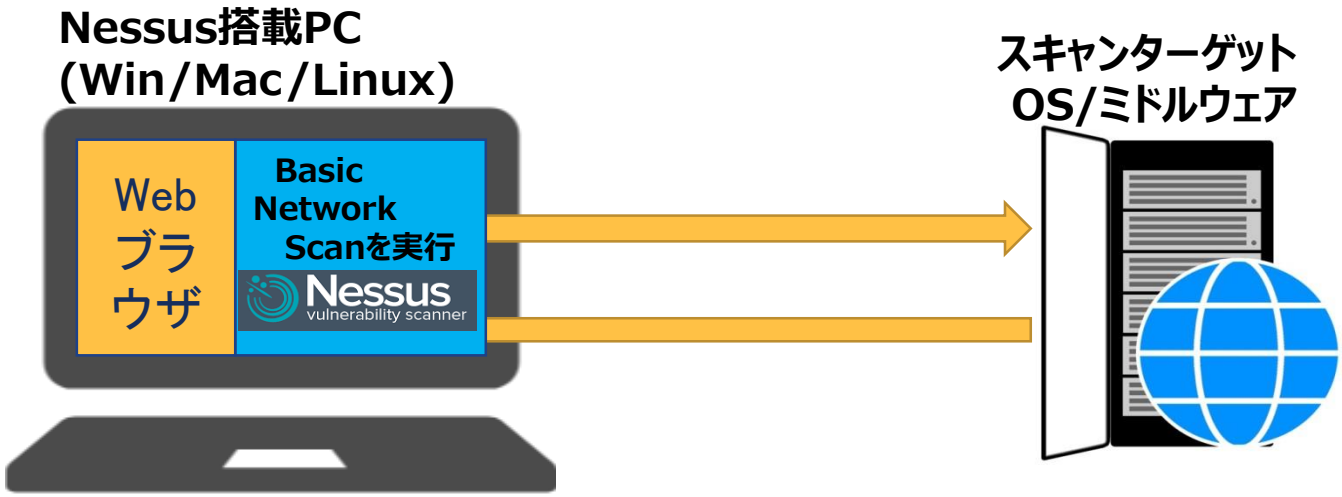
②空きポート情報が表示される

```
$ nmap -A 172.20.10.4
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-01 10:30 JST
Nmap scan report for 172.20.10.4
Host is up (0.028s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE        VERSION
135/tcp  open  msrpc          Microsoft Windows RPC
139/tcp  open  netbios-ssn    Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds   Windows 10 Home 14393 microsoft-ds (workgroup:
WORKGROUP)
5357/tcp open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
8080/tcp open  http-proxy
| fingerprint-strings:
|      ・・・・・
```

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

２）OS/ミドルウェアの脆弱性の自動診断（Nessus）

- **Nessus professionalは、サーバやネットワーク機器の脆弱性の検出が可能**
- **サーバの使用するソフトウェアの既知の脆弱性の有無、設定ミス、脆弱なPW等**
- **の確認も可能**
- **Nessusはサービスプロセスとして実行され、クライアントはブラウザを利用**



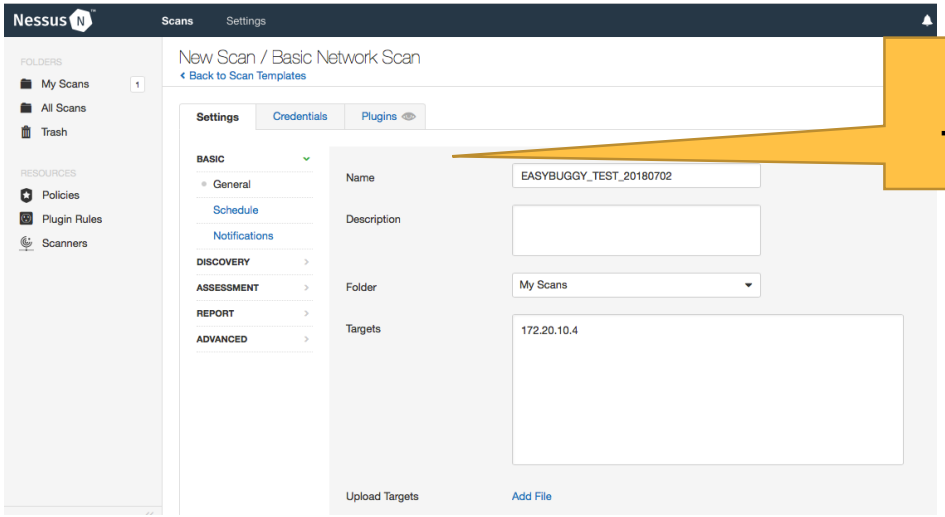
※開発元へアクセスし、最新の脆弱性DBや必要なライブラリへ更新が必要。
※無償の個人利用可能なライセンスはあるが、商用利用は不可。

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

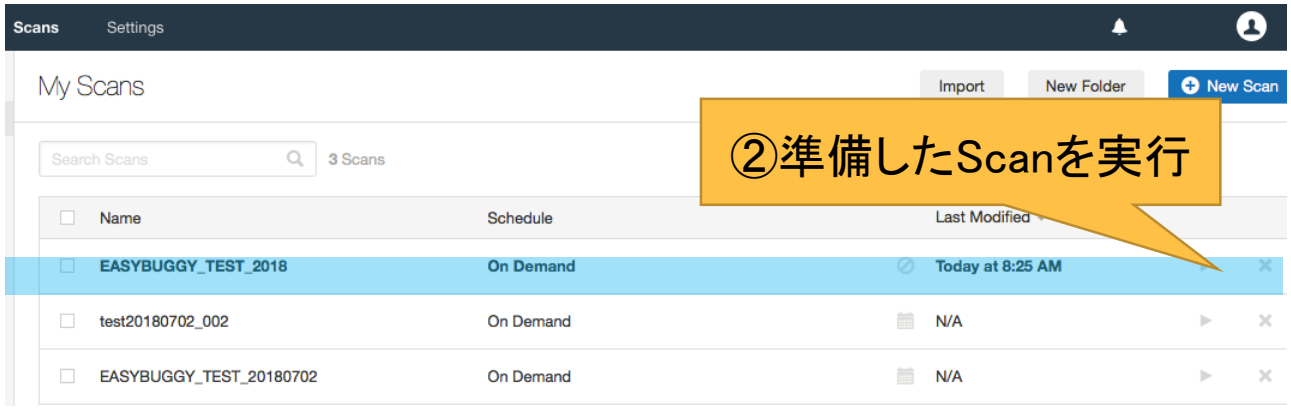
4．実践編（5）

2）OS/ミドルウェアの脆弱性の自動診断（Nessus）

①テンプレートからBasic Network Scanを選択し、設定



②保存したScanの条件を実行

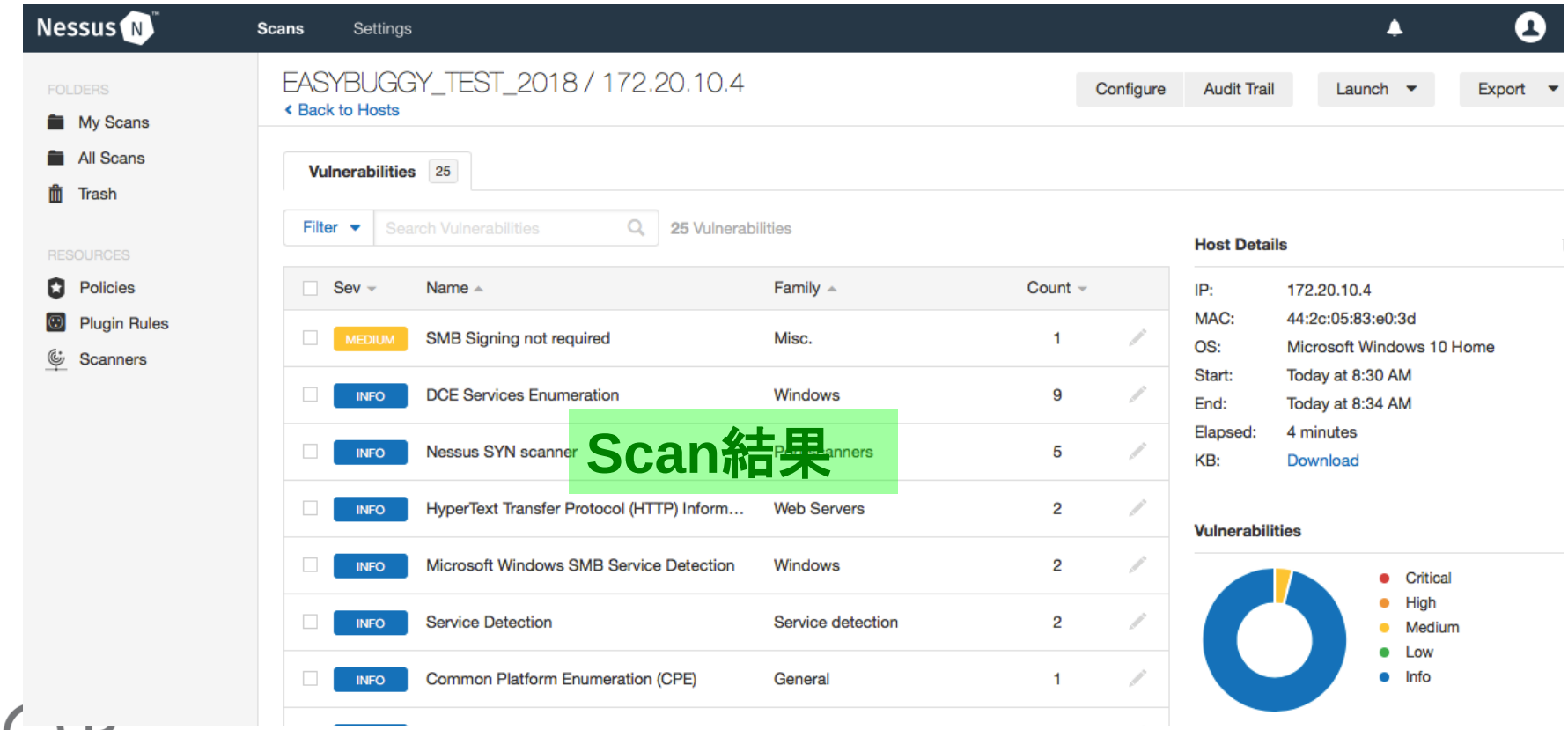


Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．実践編（6）

2）OS/ミドルウェアの脆弱性の自動診断（Nessus）結果

③Scan結果が表示される

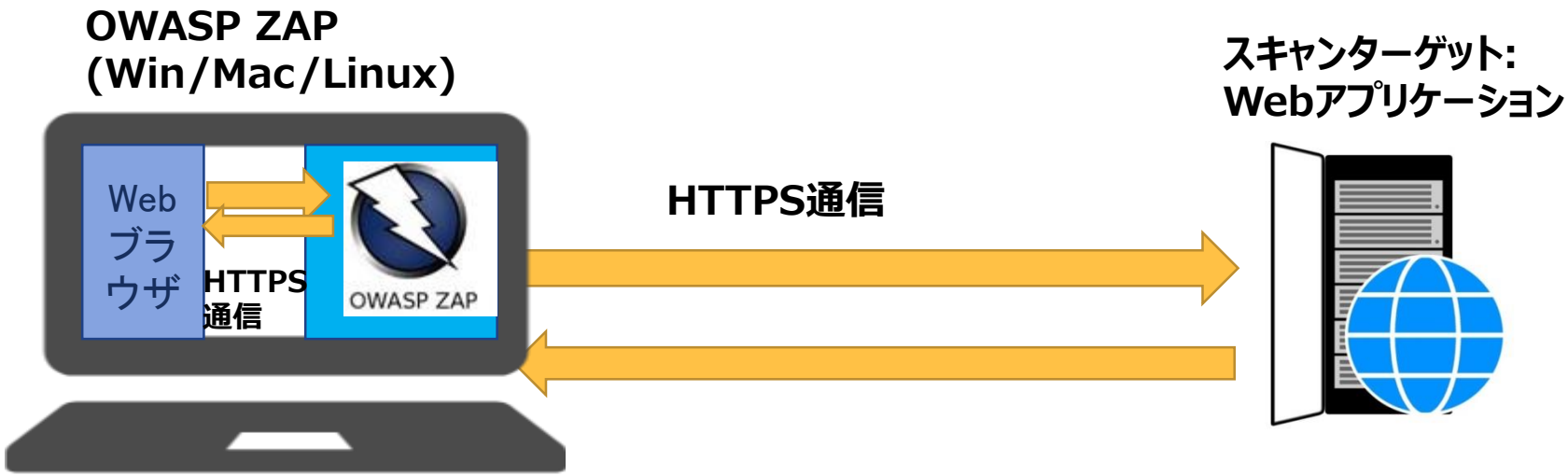


Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

４．実践編（7）

３）Webアプリの脆弱性の自動診断（OWASP ZAP）

- **OWASP ZAPは自動診断型のWebアプリケーション向け脆弱性テストツール**
- **起点となるURLを指定し、サイト全体に対して主要な脆弱性を幅広く診断可能**



※自サイトでも誤検知やサーバ障害につながりえる為、サーバ管理者と事前調整が重要。
※AWS等クラウドサービス利用の場合は、サービス提供会社へ事前申請必要。
※OWASP ZAPには攻撃モードがあり、アクセスしたWEBサイトに無差別に攻撃を仕掛ける。不用意にONにすると不正アクセスになり、法を犯す可能性あり。

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

４．実践編（8）

３）Webアプリの脆弱性の自動診断（OWASP ZAP）

OWASP ZAPは、権限設定の不備など自動脆弱性診断ツールでは検出できない脆弱性を除いて、Webアプリケーションにおける主要な脆弱性を網羅的に検査できる。

- クロスサイトスクリプティング
- SQLインジェクション
- パストラバーサル
- オープンリダイレクタ
- ヘッダインジェクション
- オートコンプリート機能の有効
- アプリケーションエラーの開示
- X-Content-Type-Optionsヘッダの未設定
- X-Frame-Optionsヘッダの未設定
- HttpOnly属性が付与されていないCookieの利用　等

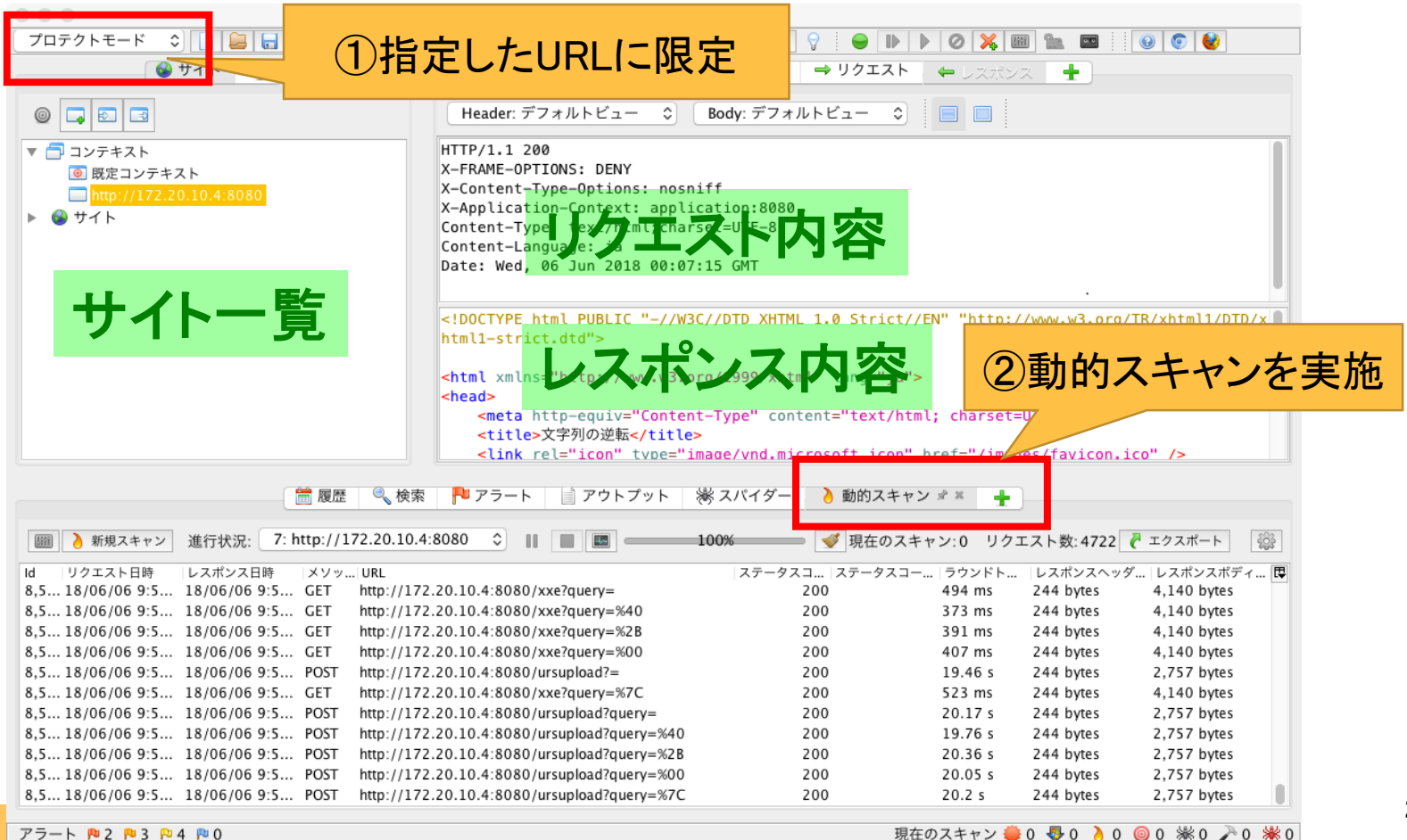
SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．実践編（9）

3）Webアプリの脆弱性の自動診断（OWASP ZAP）

OWASP ZAPをProxy設定したブラウザで、やられサーバを事前に操作。
指定したURLを指定し、動的スキャンを実行。



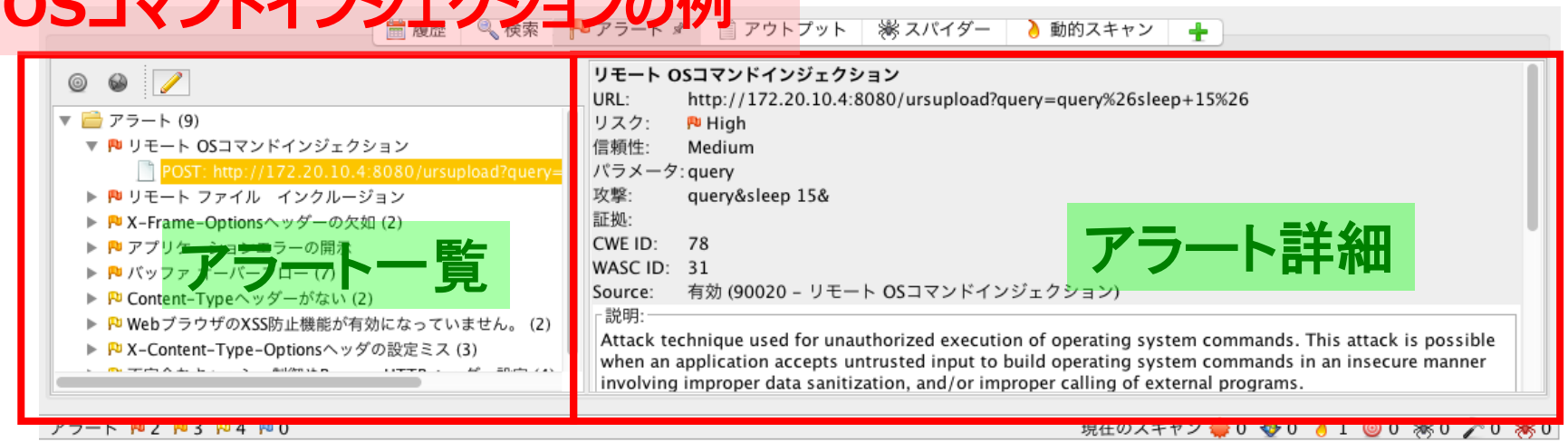
4．実践編（10）

3） Webアプリの脆弱性の自動診断（OWASP ZAP）

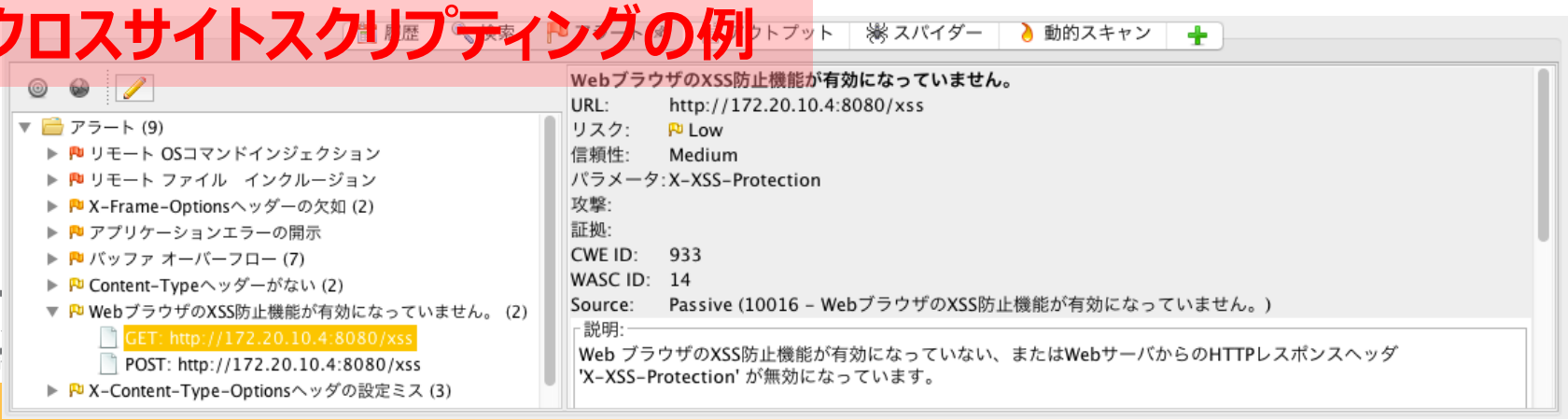
結果

やられサーバに対して動的スキャンすると、脆弱性がアラートとして出力される。

OSコマンドインジェクションの例

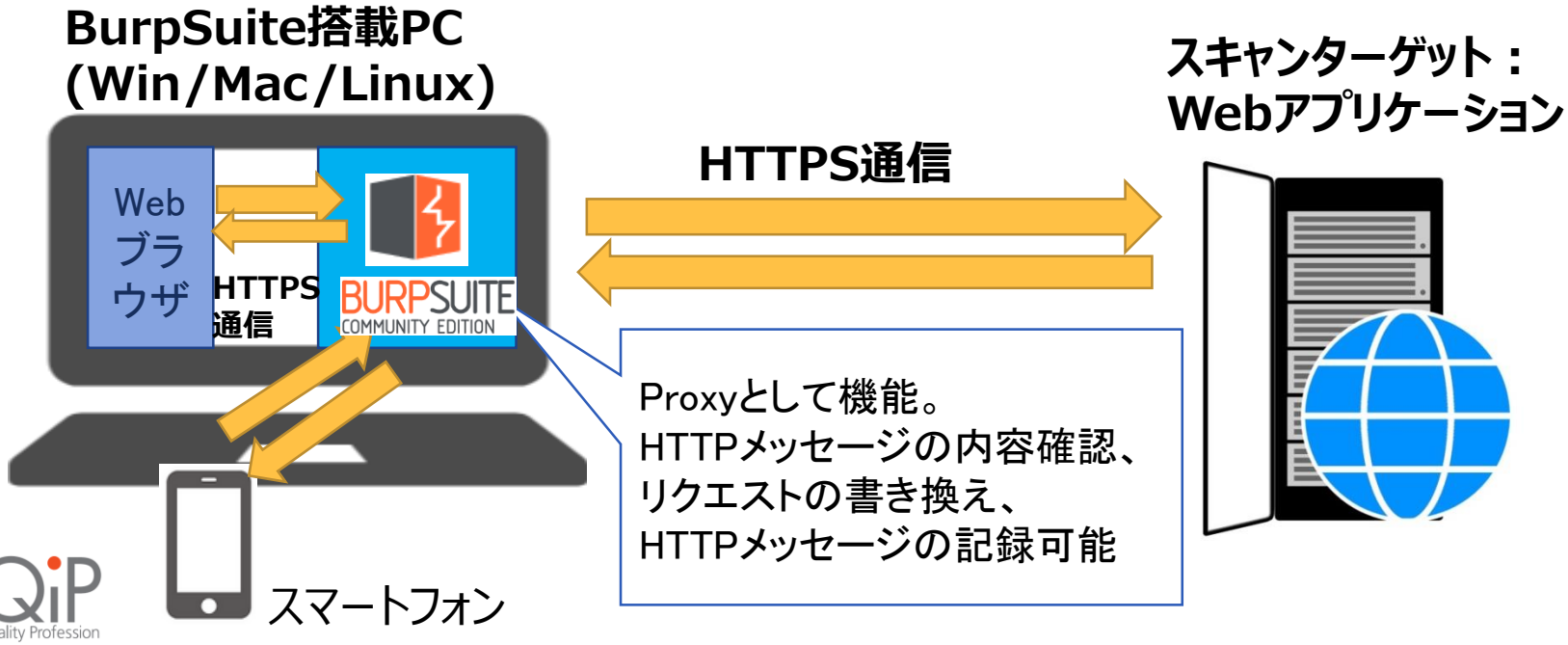


クロスサイトスクリプティングの例



4．Proxyツールを用いた手動診断（Burp Suite）

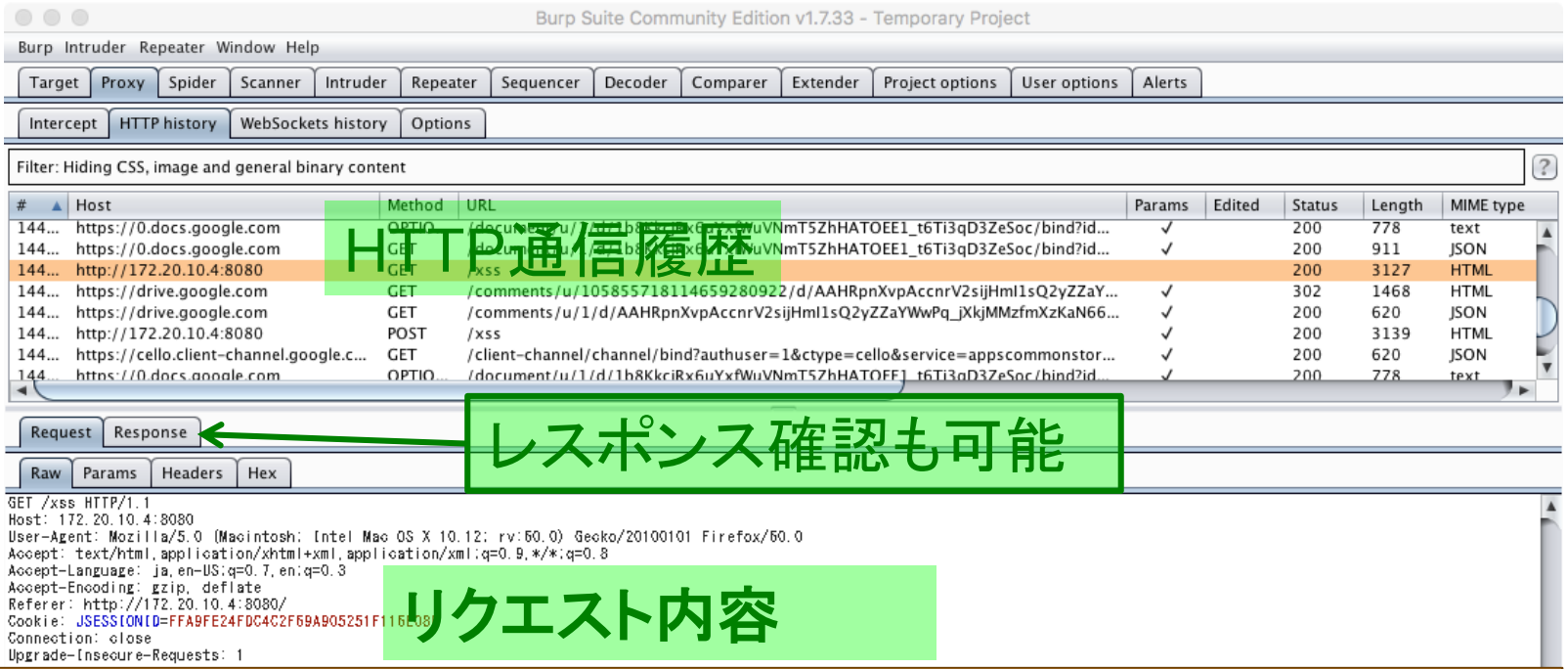
Burp Suite Community Editionは、Proxyサーバとして機能し、
WebアプリケーションへのHTTPリクエストを改ざんして、診断が可能。
※スマートフォンアプリのプロキシとしても機能し、スマートフォン側
セキュリティ診断にも利用可能。



Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．Proxyツールを用いた手動診断（Burp Suite)

Burp SuiteのProxy機能では、Webアプリケーションへの HTTP通信のリクエストやレスポンスを確認することが可能。情報を改ざんして、手動診断に利用。



※BurpSuiteのSSLサーバ証明書をクライアント（ブラウザやスマートフォン等）にインストールすることにより、HTTPSの確認も可能。
※クライアントがSSLサーバ証明書の正当性確認をしている場合は利用できない為、その場合は、一時的に外してチェックすることが必要。

4．Proxyツールを用いた手動診断（Burp Suite）

セキュリティチェックリストを実践。（SQLインジェクションの診断例）

脆弱性	**SQLインジェクション**
診断方法	**HTTPリクエストのパラメータにSQLの文字列を追加し、送信。 追加した文字列によって、HTTPレスポンスの違いがあれば、 SQL文がコードとして機能しており、脆弱性のリスク有。**
対象画面	**「暗証番号検索」**
対象パラメータ	**「name」**
追加する文字列	**[A]「' and ‘a'=‘a」 [B]「 ' and ‘a'=‘b」**
脆弱性がある場合	**[A]と[B]のレスポンスが異なる [A] は「‘a’=‘a’」がtrueで、レスポンスは正常値と同じ [B] は「‘a’=‘b’」がfalseで、検索結果が0件になる**

SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．Proxyツールを用いた手動診断（Burp Suite)

やられサーバWEBページ（EasyBuggy)

暗証番号検索

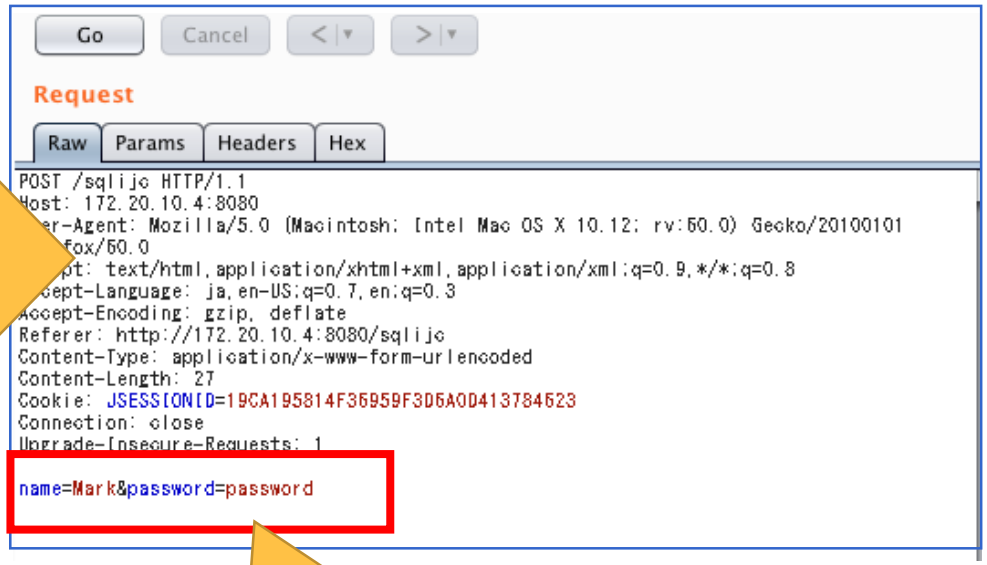
名前とパスワードを入力すると、暗証番号が表示されます。

名前： パスワード：

送信

名前	暗証番号
Mark	0000000000

Burp Suiteで取得したリクエストを改竄し、送信



Requestのパラメータを書き換えて送信

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

4．Proxyツールを用いた手動診断（Burp Suite）

結果

「正常値」

リクエスト

```
name=Mark&password=password
```

レスポンス（正常な結果）

```
<tr>
    <th rowspan="1" colspan="1">名前</th>
    <th rowspan="1" colspan="1">暗証番号</th>
</tr>
<tr>
    <td rowspan="1" colspan="1">Mark</td>
    <td rowspan="1" colspan="1">0000000000</td>
</tr>
```

[A]「‘ and ’a‘=’a」を追加

リクエスト

```
name=Mark' and' a' =' a&password=password
```

レスポンス（「正常値」と同じ結果）

```
<tr>
    <th rowspan="1" colspan="1">名前</th>
    <th rowspan="1" colspan="1">暗証番号</th>
</tr>
<tr>
    <td rowspan="1" colspan="1">Mark</td>
    <td rowspan="1" colspan="1">0000000000</td>
</tr>
```

[B]「‘ and ’a‘=’b」を追加

リクエスト

```
name=Mark' and' a' =' b&password=password
```

レスポンス（[A]と違う結果）

```
<div class="alert alert-danger" role="alert">
  <span class="glyphicon glyphicon-warning-sign"></span>
  ユーザーが存在しないか、パスワードが一致しません。
</div>
<div class="alert alert-info" role="alert">
  <span class="glyphicon glyphicon-info-sign"></span>
```

SQLインジェクションの脆弱性がありえると判定

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

5．まとめ（1）

■ 成果

■品質保証部門でも標準的な脆弱性診断が可能なことを確認

評価・テストフェーズにて、自動診断ツール及び手動診断を行い、
標準的な脆弱性の有無を評価できることを確認。

■業務の上流工程から評価・テストフェーズにおいて実利用

一部のメンバーは実際の業務にて脆弱性診断を実施。
結果、「脆弱性診断のエキスパート」に頼る前段階のレベルを上げ、
プロダクトのセキュリティ品質の底上げにつなげることも確認。

SQiP
Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

5．まとめ（2）

■ 全社的なセキュリティ対策が必要な時代に

■全社的なセキュリティ対策が必要対応

・経営者から開発、品質保証まで、全社的なセキュリティ対策が必要。

■最低限の脆弱性評価は自前で、かつ継続的に

・攻撃者は防備の弱い処から狙う。最低限の脆弱性評価は運用開始後も含め、継続的に自分たちでできることが望ましい。

・限られた「脆弱性診断のエキスパート」に難しい脆弱性評価等、優先度の高い作業に注力してもらう。

SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

5．まとめ（3）

■ 全社的なセキュリティ対策が必要な時代に

■セキュリティは今後QAエンジニアに求められるスキル

・品質保証部門はセキュリティ対策の妥当性を確認する為に、
評価基準やリテラシー、スキルが必要。

・QAエンジニアは各種ガイドラインや複数のツールを駆使した
脆弱性テストのスキルを磨くことが望まれる。

・組織はその人材を育てる覚悟が必要になる。

一緒にセキュリティ品質を
作りこみましょう！

Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

参考情報・参照

- **OWASP Top10 2017　最も重大なウェブアプリケーションリスクトップ10**
  https://www.owasp.org/images/2/23/OWASP_Top_10-2017%28ja%29.pdf
- **Microsoft SDL の簡単な実装**
  https://www.microsoft.com/ja-jp/download/details.aspx?id=12379
- **IPA　安全なウェブサイトの作り方**
  https://www.ipa.go.jp/security/vuln/websecurity.html
  - 「IPA ウェブ健康診断仕様」を利用する
    ウェブ健康診断仕様(安全なウェブサイトの作り方別冊).pdf
- **Webアプリケーション脆弱性対策チェックリスト**
  https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents2/checklist.pdf
- **Webセキュリティ担当者のための脆弱性診断スタートガイド**
  上野宣が教える情報漏えいを防ぐ技術　(翔泳社)
- **Proxy Tool (Burp Suite　Community Edition)**
  https://portswigger.net/burp/communitydownload
- **ポートスキャナ　Nmap（Network Mapper）**
  https://nmap.org/
- **脆弱性スキャナ　OWASP ZAP**
  https://www.owasp.org/index.php/Main_Page
- **脆弱性スキャナ　Nessus Professional**
  https://www.tenable.com/plugins/

SQiP Software Quality Profession

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.

End of Slide

ご清聴ありがとうございました

Copyright © 2018 ソフトウェア品質保証部長の会 All Rights Reserved.