脆弱性診断環境におけるクラウドの登場と分科会での診断技術の標準化の取り組み

Network Security Forum 2024
ISOG-J WG1 新技術に対する診断手法分科会
株式会社Flatt Security 齋藤 徳秀

自己紹介

株式会社 Flatt Security

プロフェッショナルサービス事業部

齋藤 徳秀

主な業務

- Webアプリケーション診断
- クラウド診断
- その他各種社内業務やリサーチ

スライドの流れ

- クラウドとセキュリティにおける潮流と背景の補足
  - 診断環境におけるパブリッククラウドの登場
  - 脅威への理解
  - 現実に発生している設定ミスなどの事例
  - 今後考えられる脅威
- パターン化される攻撃手法
- 分科会で整理した診断手法についての例
  - 代表的なサービスを中心にした発表

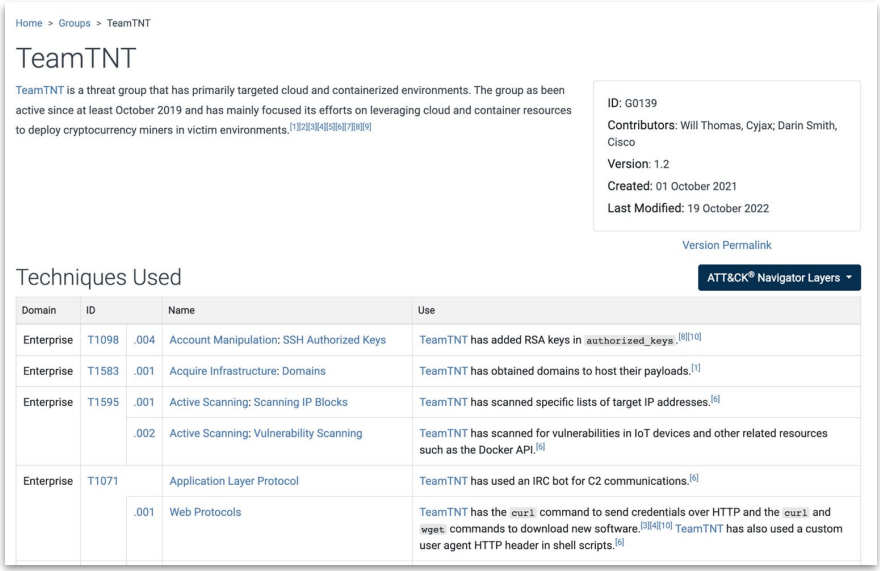
背景

background: 診断環境におけるパブリッククラウドの登場

- Webアプリケーションの近代化により、各種技術スタックも変化を続けており、インフラの構成についてもオンプレからクラウドへと変化しています。
- そのような環境下でWebやモバイル等の脆弱性診断においても、開発現場の技術変革に伴い、環境の変化が発生しています。
  - 例
    - ディスクストレージから外部のオブジェクトストレージサービスへ
    - アプリに実装された認証やユーザー管理機能からIdP サービスへ
    - オンプレからIaaS、そしてPaaSやCaaS(container as a service)、FaaS(function as a service)へ
- クラウドサービス間での操作を行う場合は、権限の定義や、それに基づき発行された認証情報を元におこないます。
  - 例:
    - IaaSから外部のオブジェクトストレージの操作を行うなどの行為

background: クラウドを標的とした脅威

現時点で、クラウドを標的とした攻撃がAPT等から、大々的に行われている報告例は少ないです。
また報告されている多くに関しては、IMDSを標的とした認証情報の取得に力を入れています。



Home > Groups > TeamTNT

TeamTNT

TeamTNT is a threat group that has primarily targeted cloud and containerized environments. The group as been active since at least October 2019 and has mainly focused its efforts on leveraging cloud and container resources to deploy cryptocurrency miners in victim environments.[1][2][3][4][5][6][7][8][9]

ID: G0139
Contributors: Will Thomas, Cyjax; Darin Smith, Cisco
Version: 1.2
Created: 01 October 2021
Last Modified: 19 October 2022

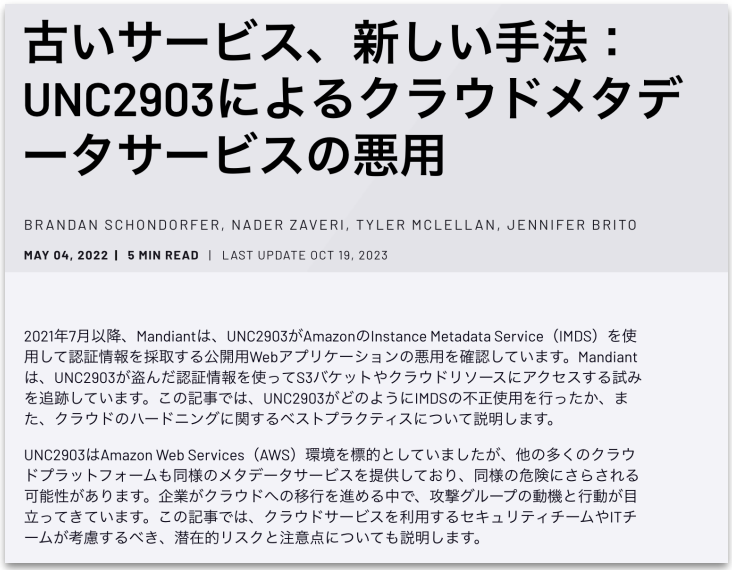
Version Permalink

Techniques Used

ATT&CK® Navigator Layers

Domain	ID		Name	Use
Enterprise	T1098	.004	Account Manipulation: SSH Authorized Keys	TeamTNT has added RSA keys in `authorized_keys`.[8][10]
Enterprise	T1583	.001	Acquire Infrastructure: Domains	TeamTNT has obtained domains to host their payloads.[1]
Enterprise	T1595	.001	Active Scanning: Scanning IP Blocks	TeamTNT has scanned specific lists of target IP addresses.[6]
		.002	Active Scanning: Vulnerability Scanning	TeamTNT has scanned for vulnerabilities in IoT devices and other related resources such as the Docker API.[6]
Enterprise	T1071		Application Layer Protocol	TeamTNT has used an IRC bot for C2 communications.[6]
		.001	Web Protocols	TeamTNT has the `curl` command to send credentials over HTTP and the `curl` and `wget` commands to download new software.[3][4][10] TeamTNT has also used a custom user agent HTTP header in shell scripts.[6]

https://attack.mitre.org/groups/G0139/



古いサービス、新しい手法：UNC2903によるクラウドメタデータサービスの悪用

BRANDAN SCHONDORFER, NADER ZAVERI, TYLER MCLELLAN, JENNIFER BRITO

MAY 04, 2022 | 5 MIN READ | LAST UPDATE OCT 19, 2023

2021年7月以降、Mandiantは、UNC2903がAmazonのInstance Metadata Service（IMDS）を使用して認証情報を採取する公開用Webアプリケーションの悪用を確認しています。Mandiantは、UNC2903が盗んだ認証情報を使ってS3バケットやクラウドリソースにアクセスする試みを追跡しています。この記事では、UNC2903がどのようにIMDSの不正使用を行ったか、また、クラウドのハードニングに関するベストプラクティスについて説明します。

UNC2903はAmazon Web Services（AWS）環境を標的としていましたが、他の多くのクラウドプラットフォームも同様のメタデータサービスを提供しており、同様の危険にさらされる可能性があります。企業がクラウドへの移行を進める中で、攻撃グループの動機と行動が目立ってきています。この記事では、クラウドサービスを利用するセキュリティチームやITチームが考慮するべき、潜在的リスクと注意点についても説明します。

https://www.mandiant.jp/resources/blog/cloud-metadata-abuse-unc2903

background: 設定ミスによる情報の漏洩の可能性

Amazon S3やGoogleの提供するFirestoreのようなサービスにおいては、設定次第でユーザーの情報が認証なしで見られる可能性があります。

Japanese medical online consultation site leaking consumer-submitted images of symptoms

Posted on March 22, 2022 by Dissent

After multiple unsuccessful attempts to get a popular Japanese medical online consultation site to secure a misconfigured bucket, researchers at SafetyDetectives have decided to publicly disclose the leak.

Doctors Me provides customers with on-demand access to professional medical advice. People can sign up for a monthly unlimited access plan (for less than $3.00 per month) or a per consultation plan with specified experts.

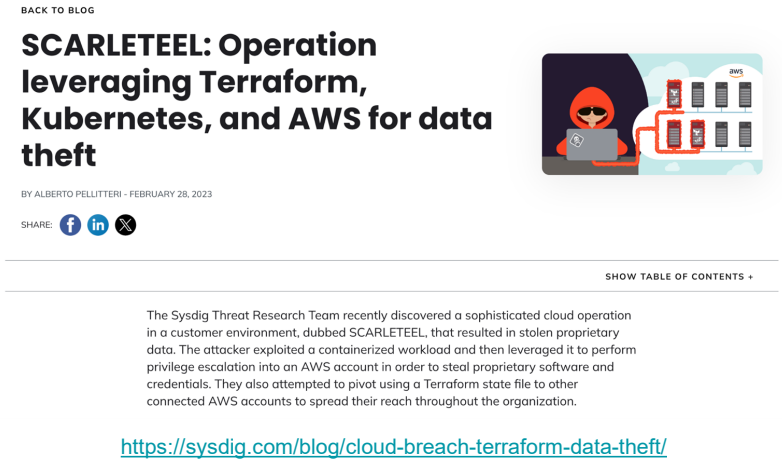
The patients can use the service anonymously, but in uploading pictures or details about themselves or their children, they may reveal identifying information. Some of the image files reportedly provide sufficient views to be able to identify some patients or children.

https://www.databreaches.net/japanese-medical-online-consultation-site-leaking-consumer-submitted-images-of-symptoms

background: 今後考えられる攻撃と備え

クラウドの利活用やクラウドネイティブというものは、登場から提唱まで一定の時間が経過し、実際の運用にたりうる状況になりつつある。その中で、企業や開発現場の技術選定においても、十分考慮され、採用に至るケースも多くなっている。その中で、今後はより多くのクラウドサービスが、プロダクトに組み込まれるようになると考えます。

そのような背景から、攻撃者もクラウド環境への慣れや標的をオンプレとクラウド双方に向ける可能性も存在します。



https://sysdig.com/blog/cloud-breach-terraform-data-theft/

background: 今後考えられる攻撃と備え

備えとして、脆弱性診断を行うセキュリティエンジニアにおいても、侵入の起点となる脆弱性や設定ミスの発見を行えるように、日々自己研鑽をする必要です。

また、事業会社においては、クラウド等の資産の把握や、保有している資産から想定される攻撃経路などを洗い出し、優先度を決めた上で、適切な対策を施す必要があると考えられます。

パターン化される攻撃手法

- 公然の知識としての設定ミス
  - 公開されたドキュメントに記載された「デフォルト設定」などを考慮しないことによって発生するアプリケーションとの不整合
  - Amazon Cognito User Poolにおける暗黙的なデフォルト値
- 非公然の設定ミス
  - 明示的に設定が変更されており、その設定値が脆弱な状態
  - クラウドサービス多用するアプリケーションにおける設計の不備
  - Amazon S3 bucketにおけるObjects Listing
  - CNAMEにクラウドサービスのドメイン・IPを指定する
- 実装によるクラウドサービスの悪用
  - アプリケーションの実装に埋め込まれた脆弱性やそれらを利用したクラウドサービスの利用
  - SSRFなどの脆弱性を用いたIMDSへのアクセス

診断の手法の例

Webアプリケーションの脆弱性を利用した認証情報の窃取

パターン化される攻撃手法の区分

実装によるクラウドサービスの悪用

概要

クラウド環境を利用してWebサービスなどを提供するシステムにおいて、アプリケーションの脆弱性を利用し、クラウドサービス側の仕様などをもとに、認証情報を窃取される可能性があります。そのため、WG1 NewTechでは、クラウドサービスの仕様などを元に、いかに脆弱性を発見できるかについて、整理をしました。

https://webapppentestguidelines.github.io/newtechtestdoc/docs/cloudsec/cloud_credential/

Webアプリケーションの脆弱性を利用した認証情報の窃取

EC2に代表されるIaaSの場合は、Identity Meta Data Service(IMDS)と呼ばれるインスタンスに関連する情報を取得可能なエンドポイント(169.254.169.254に代表されるIPなど)へアクセスすることで、IAMから払い出された認証情報を取得しています。他にもAWS Lambdaの場合は、認証情報を環境変数に保存している特性があるなど、各サービスによって異なるものの、脆弱性を用いてこれら特定のファイルなどにアクセスすることで認証情報を取得することが可能になります。

例:

- SSRFを用いたIMDSへのアクセス
- OS Command Injectionを用いた環境変数等へのアクセス
- Path Traversalを用いた/proc/self/environなどへのアクセス

Webアプリケーションの脆弱性を利用した認証情報の窃取

正常遷移のリクエスト

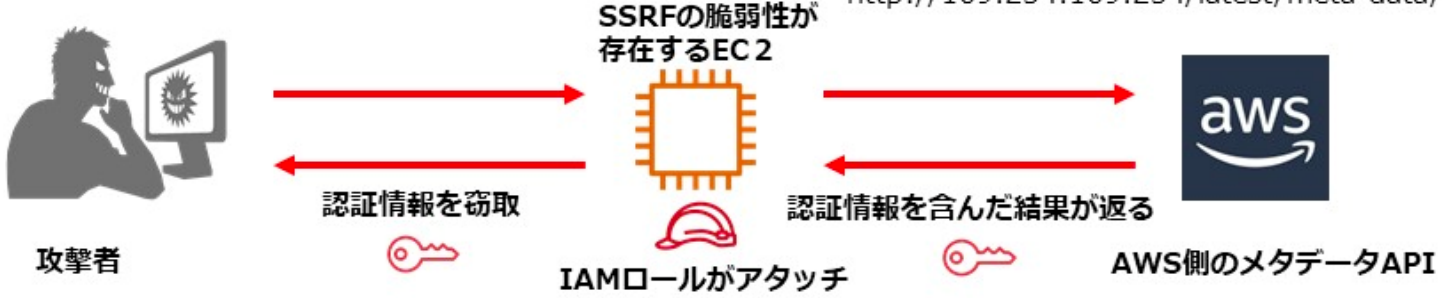
xxx/?url=http://www.example.com

①パラメータの値をメタデータサーバのURL値に変更して送信

xx/?url=http://169.254.169.254/latest/meta-data/…

②EC2からAWSのメタデータAPIに対して一時的な認証情報の取得を勝手に要求される

http://169.254.169.254/latest/meta-data/…



③攻撃者は脆弱性が存在するEC2からのレスポンスから、EC2に一時的に付与された認証情報（アタッチされたIAMロールに紐づく）を取得することができる

Webアプリケーションの脆弱性を利用した認証情報の窃取

対策としては、従来の脆弱性への対策と共に、提唱される緩和策や、認証情報のポリシーを最小権限の原則に基づき設定することで、漏洩した場合のリスクを軽減することが可能になります。

例:

- AWSの場合IMDSv1をIMDSv2へ変更することで、GETのみのSSRFではアクセスできなくなるため、緩和策として利用可能
- LambdaやEC2へ付与するIAMが利用される場所を限定して、外部からの利用や、異なるアカウントからの利用をできないようにする
    - IAM ポリシーのリソースやConditionのPrincipalAccountなどを用いて制約を設定する

意図しないサインアップ経路の存在

パターン化される攻撃手法の区分

公然の知識としての設定ミス

概要

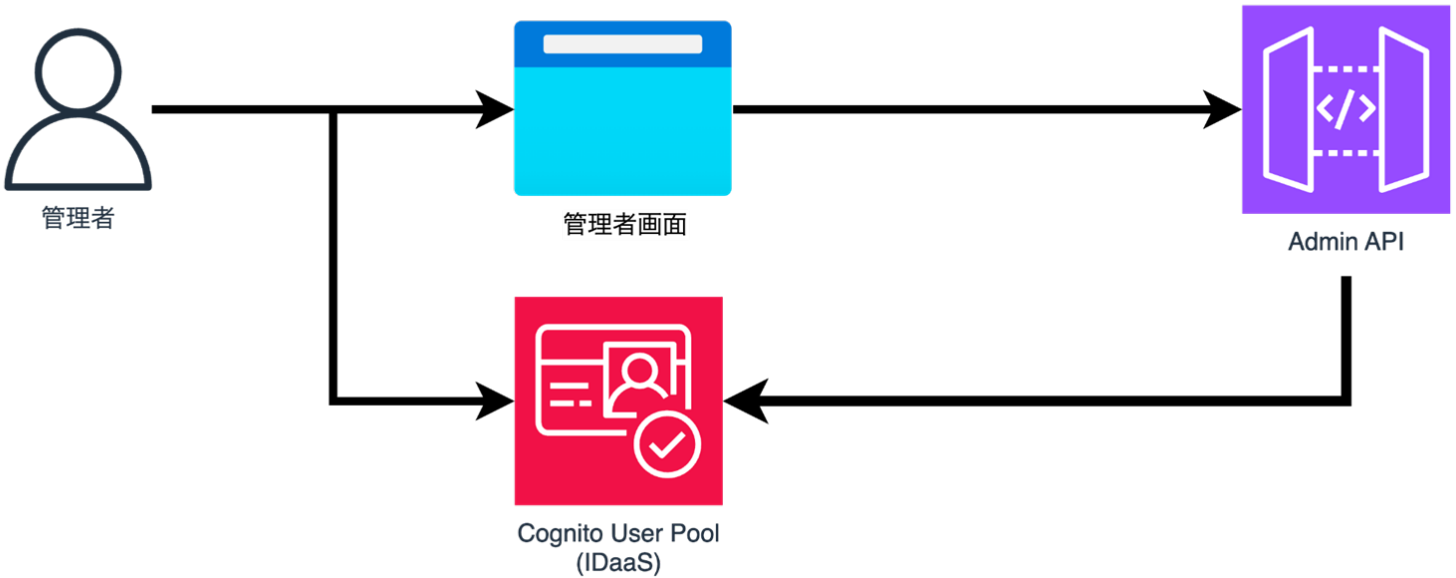
パブリッククラウドでは、ある機能に特化したサービスというものが多く存在します。その中でもWebサービスに欠かせない認証機能においても、サービスとして存在しており、それらをIDaaS(IdP)と呼びます。

このようなサービスにおいて、多くのユースケースに対応するために、自らユーザー登録を行えるようにする機能が提供されているケースが多数です。このような機能はECサイトなどの公に公開されたWebサービスであれば、特に問題にはなりませんが、これが管理者画面などの限られた利用者しかいないアプリケーションでは話が別です。

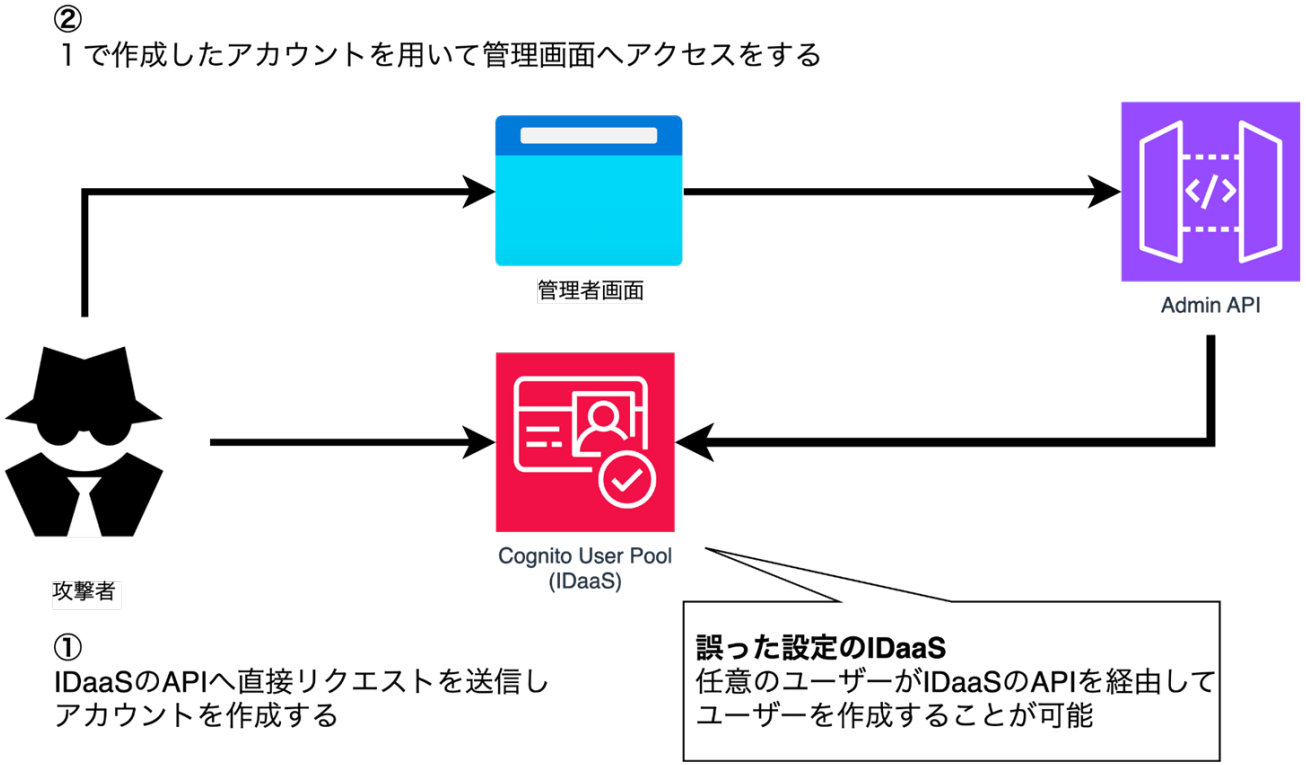
https://webapppentestguidelines.github.io/newtechtestdoc/docs/cloudsec/idaas/presence_of_unintended_signup_paths/

意図しないサインアップ経路の存在 - 構成やユースケース

正常なリクエスト
管理者がAdmin APIを経由して管理者を作成する
APIを経由して作成されたアカウントを用いてIDaaS
でログインをする



意図しないサインアップ経路の存在 - 攻撃の手法と診断へ



意図しないサインアップ経路の存在

対策としては、限定されたサインアップの場合、自己サインアップをはじめとした、利用者が自らユーザを作成できる機能の無効化してください。また、自己サインアップ機能を無効化できない場合、それに準ずる対策が必要となります。

自己サインアップを許可しない設計や要件の場合、根本的対策として該当する機能を無効化し、ユーザによるサインアップを許可しないようにしてください。 無効後に新規作成する際は、RBAC等のアクセス制御を用いて、操作が可能なユーザを限定してください。

自己サインアップを無効化できない場合は、管理者による新規作成フローにおいて、利用者から操作できないカスタム属性などを用いて該当アカウントが管理者作成のアカウントであることを証明する属性を追加し、認可やアクセス制御の際に確認してください。

まとめ

まとめ

- クラウドサービスに対しての組織的な脅威については現時点では多くは見られない。一方で、個人や特定の目的を持った犯罪者が、クラウドサービスの認証情報を標的に攻撃をするケースは報告されている。
- サービスの提供環境の変遷に伴い、今後攻撃者においても標的にする「モチベーション」も増加してきている
- このような背景から、ISOG-J WG1 Newtech分科会では、クラウドサービスでの脆弱性をいかに見つけ出すのかについてまとめました
- 実環境で発生する脆弱性のパターンを元に、現在利用が多いサービスを中心に診断手法を公開しました。

まとめ - パターン化される攻撃手法

- 公然の知識としての設定ミス
  - 公開されたドキュメントに記載された「デフォルト設定」などを考慮しないことによって発生するアプリケーションとの不整合
  - Amazon Cognito User Poolにおける暗黙的なデフォルト値
- 非公然の設定ミス
  - 明示的に設定が変更されており、その設定値が脆弱な状態
  - クラウドサービス多用するアプリケーションにおける設計の不備
  - Amazon S3 bucketにおけるObjects Listing
  - CNAMEにクラウドサービスのドメイン・IPを指定する
- 実装によるクラウドサービスの悪用
  - アプリケーションの実装に埋め込まれた脆弱性やそれらを利用したクラウドサービスの利用
  - SSRFなどの脆弱性を用いたIMDSへのアクセス

ISOG-J WG1
新技術に対する診断手法分科会
運営と診断技術のご紹介

三井物産セキュアディレクション株式会社　廣田一貴
三井物産セキュアディレクション株式会社　山本健太
株式会社 Flatt Security　齋藤 徳秀

日本セキュリティオペレーション事業者協議会

Agenda

運営

- 20-30代の非営利コミュニティ運営
- 継続のための仕組み

診断手法紹介

- WebCache
- クラウド関連

自己紹介

三井物産セキュアディレクション株式会社
診断サービスチーム・レッドチーム

廣田　一貴

業務：Web脆弱性診断・ペネトレ

主な社外活動：
ISOG-J WG1 NewTech　リーダー
セキュリティキャンプ　地域連携Gr（ミニキャンプの企画）

ISOG-J/WG1

日本セキュリティオペレーション事業者協議会（ISOG-J）

- セキュリティオペレーションサービスの普及
- サービスレベルの向上

セキュリティオペレーションガイドラインWG(WG1)

- 主に脆弱性診断に関するドキュメントを作成
- To診断員、To開発者、To診断発注企業　等様々

【WG1】セキュリティオペレーションガイドラインWG（2012年7月発足）

脆弱性診断事業者・脆弱性診断士から開発会社向けまでセキュリティ技術の向上に役立つガイドライン作成を主目的としたWGです。

WGリーダー

上野 宣　株式会社トライコーダ

成果物	Webシステム／Webアプリケーションセキュリティ要件書 Webアプリケーション脆弱性診断ガイドライン 脆弱性診断士（Webアプリケーション）スキルマップ＆シラバス 脆弱性診断士（プラットフォーム）スキルマップ＆シラバス 脆弱性診断士倫理綱領 GraphQL 脆弱性診断ガイドライン ペネトレーションテストについて 脆弱性情報開示のためのチートシート その他、アジャイル開発におけるセキュリティなどに取り組んでいる
検討テーマ	要求にマッチしたセキュリティ診断サービスを的確に効率よく選択できるように、ユーザ向けセキュリティ診断サービスの解説書を作成する。セキュリティ診断サービスを向上するために、サービスを提供している技術者のレベルを計ることが可能な指標について検討する。
リーダーの思い	本WGではWG参加者同士が積極的に情報交換をする場を提供したいと考えています。ご自身の経験や知見を活かしてみたい方のご参加お待ち致しております。

新技術に関する診断手法検討部会

WG1内の部会 通称NewTech	新技術に関する 調査・研究	20～30歳前後 紹介制

→ 非営利かつ若手？の技術コミュニティ

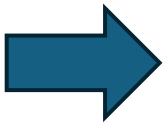
何故産まれたか

構成の変容

- Slack総数80名
- 平均年齢+1歳化

本職の方向転換

- Penシフト
- 経営シフト



若手に任せる

- リーダーを指名
- 後はリーダーが運営する
- メンバーもリーダー次第

何故産まれたか

構成の変容

- Slack総数80名
- 平均年齢+1歳化

本職の方向転換

- Penシフト
- 経営シフト

若手に任せる

- リーダーを指名
- 後はリーダーが運営する
- メンバーもリーダー次第

丸投げ

継続の仕組み

面白そう

楽しそう

モチベーション

やりたい　けど

できない

成果物へのシステム

オンライン

オフライン

コミュニケーション

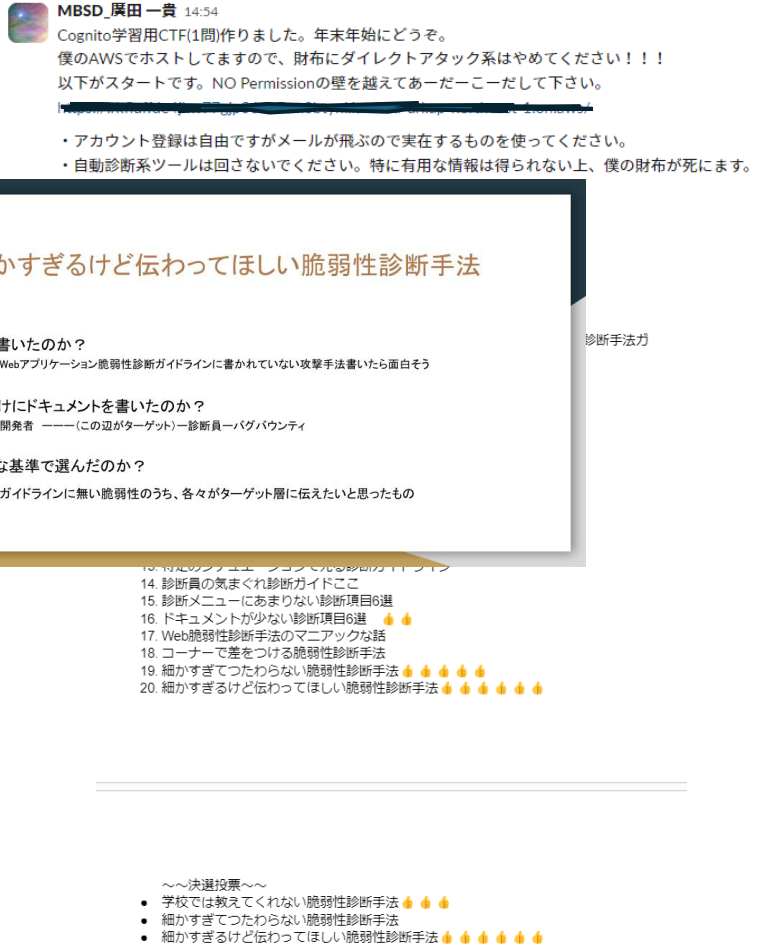
面白そう
楽しそう

面白そう

- まだ触ったことが無い
- 知らなかった
- “比較的新しい”ならOK

楽しそう

- まだ良い資料が無い
- ニーズがあるか不明
- "エンジニア"から見て俺得/私得ならOK



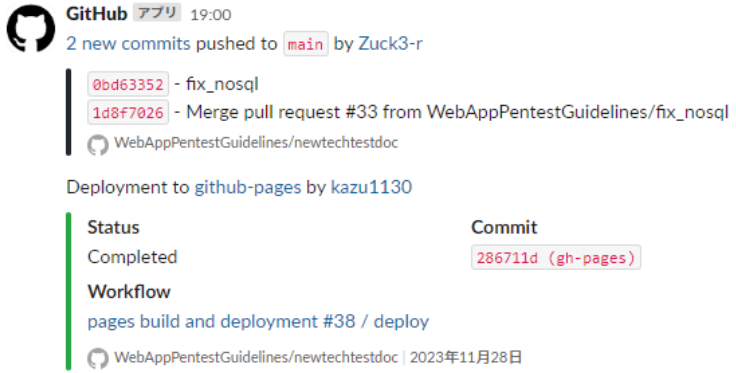
やりたい
けど
できない

- 強制的に時間をとることで進捗を作る
  - 月1回1時間MTG
  - タスクと締め切りを小さく短く

- 忙しい時はSlackに進捗書込
  - 他のイベント等が被っている時期
  - 繁忙期（まさに今）

- 自動化できるところは自動化
  - ビルドやLintの自動化
  - Issue通知等のSlack連携

```yaml
name: textlint
'on':
  - push
  - pull_request
jobs:
  textlint:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - run: >-
          npm install textlint textlint-rule-prh textlint-rule-preset-ja-technical-writing
      - run: npx textlint "content/**/*.md"
```



オンライン
オフライン

オンライン

- 東京にいないメンバー
- ほぼ全員リモートワーク
- Skype+GoogleDrive議事録

オフライン

- 飲み食いと共に実施
- 新年会・リリース記念など

現状と課題

- 企画〜作成までは順調
  - 定期的にドキュメントを出せるようになっている
  - スケールしないが十分
  - 興味ベースの企画なので完成まで持っていきやすい

- レビューや保守が課題
  - あまりシステム化出来ていない
  - 何を/どこまでチェックするか等が曖昧
  - 自動化できるところは自動化を進めていく

細かすぎるけど伝わってほしい脆弱性診断手法ドキュメント

https://webapppentestguidelines.github.io/newtechtestdoc/

Web Cache Poisoning

Network Security Forum 2024
三井物産セキュアディレクション株式会社
山本 健太

自己紹介

三井物産セキュアディレクション株式会社
診断サービス・Webアプリケーション診断

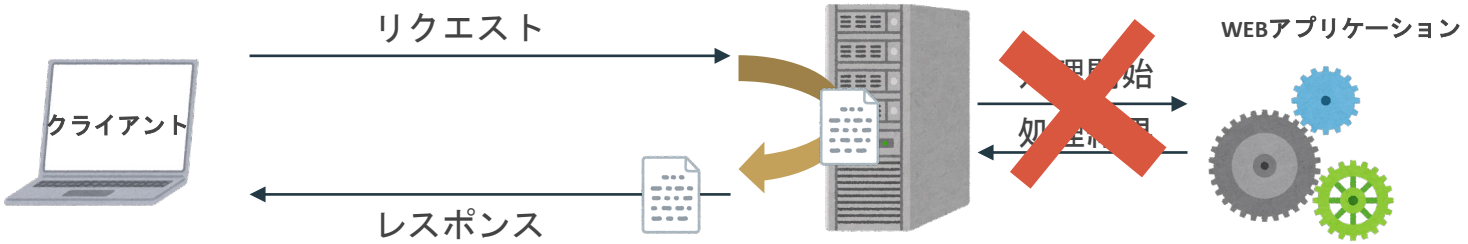
山本 健太

主な業務

- Webアプリケーション診断
- 教育
- セキュア設計支援

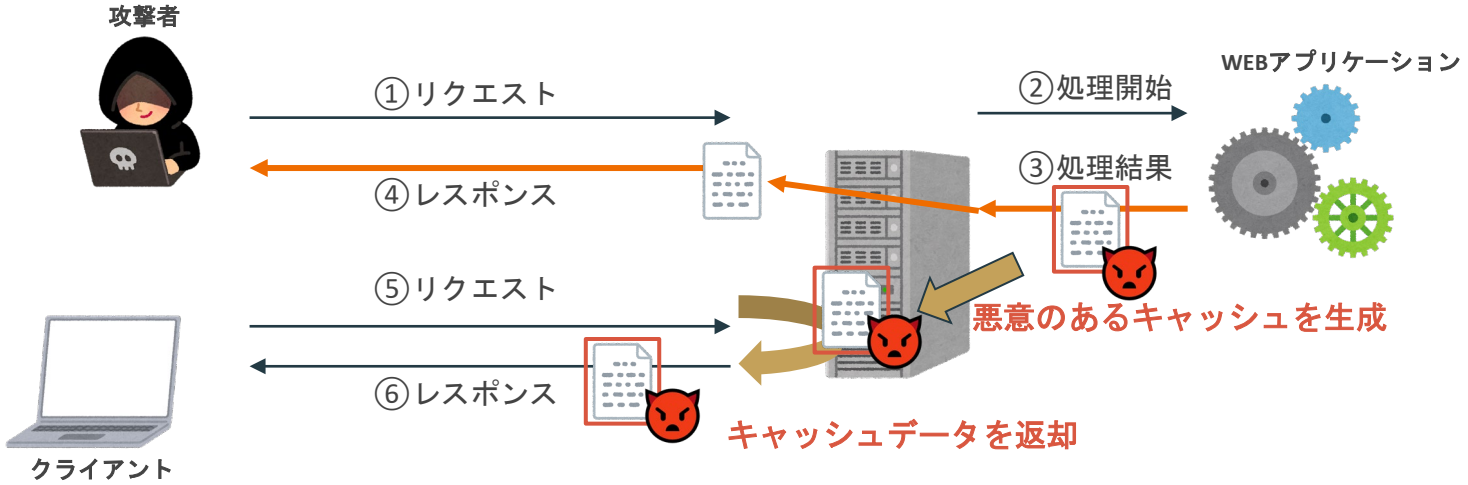
リッチコンテンツ時代の効率的なコンテンツ配信が必要

- 繰り返し利用される静的コンテンツはWebアプリケーションの負荷軽減のため事前に生成したファイルを配信する仕組みとしてキャッシュが採用される
- 更にリッチなコンテンツ、グローバル化によってCDNが広まりキャッシュの重要性が変化している



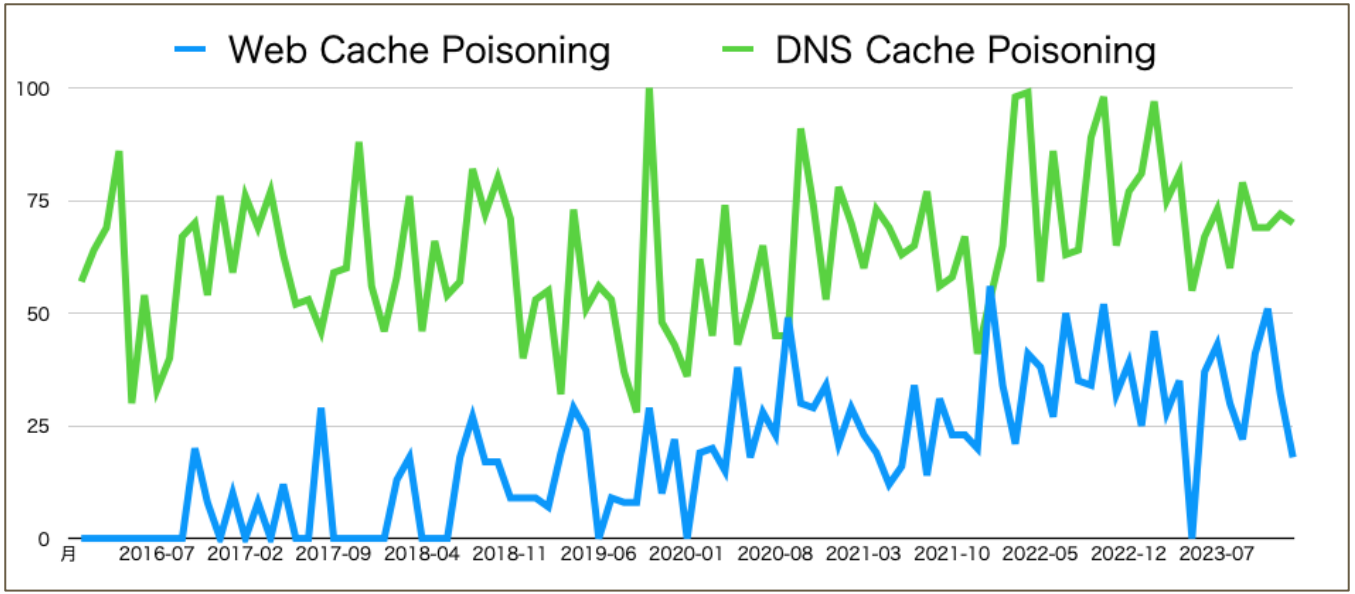
Web Cache Poisoning とは

攻撃者が不正なデータを含んだキャッシュを生成させるよう仕向けて様々な悪用を行う脆弱性の総称



Web Cache Poisoning とは

CDNの台頭などよりキャッシュが重要になる潮流の中でも、Web Cache Poisoning はメジャーな脆弱性より名前も診断方法もあまり知られていない



Google Trends 2015〜2024

原因

- 動的機能でのキャッシュ設定が不適切
- 動的機能での悪用可能な挙動が存在する

影響

Web Cache Poisoning は本来、悪意のあるコンテンツを配信させる手法に当たるため、影響は配信させるコンテンツの内容による

- XSS (クロスサイトスクリプティング)
- オープンリダイレクト
- DoS
- etc…

実例

- 不正なリソース読み込み先をキャッシュさせ、キャッシュが有効な間はWebサイトを利用できなくなる（DoS）
- 多くの報告ではXSSは再現できておらず、懸念を記載するにとどまっている
- Web Cache Poisoning ではないが、キャッシュ設定のミスにより起こったインシデントなどは少数ながらも報告されている

診断方法

前提概念

キーあり入力

キャッシュサーバがコンテンツの同一性を解釈するパラメータやヘッダのこと（？？）

キーなし入力

キャッシュサーバが解釈しないパラメータやヘッダのこと（？？？）

前提概念

```
GET /search?query=books HTTP/1.1
Host: example.com
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: ja,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
X-Forwarded-For: https://10.0.0.1:8080
```

前提概念

キー**あり**入力のケースが多い

```
GET /search?query=books HTTP/1.1
Host: example.com
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: ja,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
X-Forwarded-For: https://10.0.0.1:8080
```

キー**なし**入力のケースが多い

診断方法(リリースドキュメントより)

1. キャッシュに影響のない「キー**なし**入力」を探す
   - ○ データをキャッシュするかどうかを判断する「キャッシュキー」ではない入力となるパラメータを探す
   - ○ サーバはその他のキャッシュキーでキャッシュするかどうかを判断するため、それに影響されない入力パラメータを探す必要があります

2. 「キー**なし**入力」を送信して、コンテンツ内に出力している箇所を探す
   - ○ キーなし入力を悪意のあるコンテンツを生成するために使用します
   - ○ キーなし入力であっても悪意のあるコンテンツが生成できなければ悪用できません

3. 「キー**なし**入力」を除去して、コンテンツ内に出力しているかを確認
   - ○ キーなし入力はキャッシュを返却するかどうかの判断には使用されないので、キーなし入力を除去しても手順2で確認したコンテンツが返却されるはずです

診断方法(リリースドキュメントより)

Details Output Errors

Output to system console

Save to file: Select file ...

Show in UI:

```
1  Using albinowaxUtils v1.03
2  This extension should be run on the latest version of Burp Suite. Using an older version of Burp may cause impaired functionality.
3  Loaded Param Miner v1.4d
4  Updating active thread pool size to 8
5  Queued 1 attacks
6  Initiating header bruteforce on 0ae40066033077d8c0442c8d00b4005f.web-security-academy.net
7  Identified parameter on 0ae40066033077d8c0442c8d00b4005f.web-security-academy.net: x-forwarded-host~%s.%h
8  Identified parameter on 0ae40066033077d8c0442c8d00b4005f.web-security-academy.net: origin~https://%s.%h
9  Identified parameter on 0ae40066033077d8c0442c8d00b4005f.web-security-academy.net: origin
10 Identified parameter on 0ae40066033077d8c0442c8d00b4005f.web-security-academy.net: x-forwarded-host
11 Completed attack on 0ae40066033077d8c0442c8d00b4005f.web-security-academy.net
12 Completed 1/1
13
```

診断方法(リリースドキュメントより)

Request

Pretty Raw Hex

```
1 GET /product?productId=1&cache=1234 HTTP/1.1
2 Host: 0acf00730338a24fc03df73e00180094.web-security-academy.net
3 Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
  like Gecko) Chrome/107.0.5304.107 Safari/537.36
8 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
  /apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
9 X-Forwarded-Host: example.com
10 Referer: https://0acf00730338a24fc03df73e00180094.web-security-academy.net/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: ja,en-US;q=0.9,en;q=0.8
13 Connection: close
14
15
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=utf-8
3 Set-Cookie: session=dBqYsKTg2qQqgEWbhoSO6diiQMhaaDXi; Secure; H
  SameSite=None
4 Cache-Control: max-age=30
5 Age: 0
6 X-Cache: miss
7 Connection: close
8 Content-Length: 4229
9
10 <!DOCTYPE html>
11 <html>
12   <head>
13     <link href=/resources/labheader/css/academyLabHeader.css re
14     <link href=/resources/css/labsEcommerce.css rel=stylesheet>
15     <title>
         Web cache poisoning with an unkeyed header
       </title>
16   </head>
17   <body>
18     <script type="text/javascript" src="//example.com/resources
       </script>
19     <script src="/resources/labheader/js/labHeader.js">
       </script>
```

診断方法(リリースドキュメントより)

Send | Cancel | < | > | Target: https://0acf00730338a24fc03df73e00180094.web

Request

Pretty | Raw | Hex

```
GET /product?productId=1&cache=1234 HTTP/1.1
Host: 0acf00730338a24fc03df73e00180094.web-security-academy.net
Sec-Ch-Ua: "Chromium";v="107", "Not=A?Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/107.0.5304.107 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: https://0acf00730338a24fc03df73e00180094.web-security-academy.net/
Accept-Encoding: gzip, deflate
Accept-Language: ja,en-US;q=0.9,en;q=0.8
Connection: close
```

X-Forwarded-Hostを消しても
キャッシュされたレスポンスが返却されている

Response

Pretty | Raw | Hex | Render

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Cache-Control: max-age=30
Age: 6
X-Cache: hit
Connection: close
Content-Length: 4229

<!DOCTYPE html>
<html>
  <head>
    <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet>
    <link href=/resources/css/labsEcommerce.css rel=stylesheet>
    <title>
      Web cache poisoning with an unkeyed header
    </title>
  </head>
  <body>
    <script type="text/javascript" src="//example.com/resources/js/tracking.js">
    </script>
    <script src="/resources/labheader/js/labHeader.js">
    </script>
    <div id="academyLabHeader">
      <section class='academyLabBanner'>
        <div class=container>
          <div class=logo>
          </div>
          <div class=title-container>
            <h2>
              Web cache poisoning with an unkeyed header
```

対策/まとめ

1. Web Cache Poisoning 不正なキャッシュを生成させ、被害者に配信する攻撃
2. 診断方法やメカニズムを解説した日本語の情報が少ない
3. 対策：キャッシュを無効化
    - 機能そのものを無効化すればWeb Cache Poisoningの脅威はなくなるが、キャッシュの恩恵はなくなる
4. 対策：キャッシュの設定を見直す
    - 悪意のあるコンテンツは動的機能によって生成される
    - 本来キャッシュは繰り返し利用される静的コンテンツに適用されるべきであり、運用はキャッシュ対象を精査するべきである
    - Webサーバだけでなく、CDNの設定を見直す。CDNのキャッシュ挙動はサービスによって異なり、解釈するヘッダも異なる