

脆弱性管理についての整理

総関西サイバーセキュリティLT大会（第49回）

2025-02-12

脆弱性対応勉強会

hogebuga

初めに

本発表では、WEBアプリケーションにおける脆弱性ではなく、ソフトウェアやプラットフォームの脆弱性に焦点を当てて話をします。

1. 脆弱性とは
2. 脆弱性の評価
3. 脆弱性の対応判断
4. 脆弱性対応を行う組織の組成
5. まとめ

hogebuga (INOUE Kei)

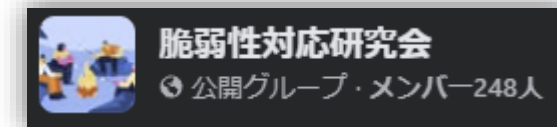
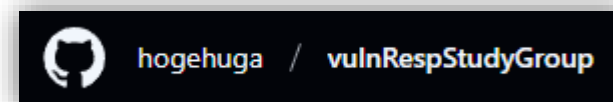
- 脆弱性対応研究会主催
 - 一応、研究会の中の勉強会が“脆弱性対応勉強会”
- 趣味
 - バイク、水風呂
- 経歴等
 - 非IT企業の情報システム部で、運用及び開発
 - MSP(Managed Service Provider)で、ユーザ企業のシステム運用や提案、構築
 - 某組合で、個人情報を含むシステムの運用、開発
 - コンサルティング会社で、セキュリティ製品エバンジェリスト、セキュリティコンサル、等
 - 最近KDDIと話題になっている会社で、脆弱性管理等の研究職
- 発表歴
 - Code Blue Open Talks(2022)
 - Internet Week2023,2024
 - Internet Week SHOWCASE IN 福岡
 - NCA Annual Conference 2023,2024
 - OWASP (Japan|Kansai|Nagoya)
 - Hardening Conference
 - 塩尻サイバーセキュリティ勉強会
 - 脆弱性対応勉強会
- その他
 - ISOG-J WG1 “脆弱性トリアージガイドライン作成の手引き”
 - ISOG-J WG1/OWASP “セキュリティエンジニアの知識地図”
 - ISOG-J WG6 “セキュリティ対応組織の教科書 第3.x版”



DALLE-Eが表現する
登壇者が後ろから刺される絵



<https://zei yakusei taiou kenkyukai.connpass.com/>



<https://www.facebook.com/groups/zei yakusei taiou kenkyukai>

脆弱性対応勉強会は個人で行っている活動であり、会社とは無関係です。
私の発言は、会社及び組織を代表する見解ではないことがあります。

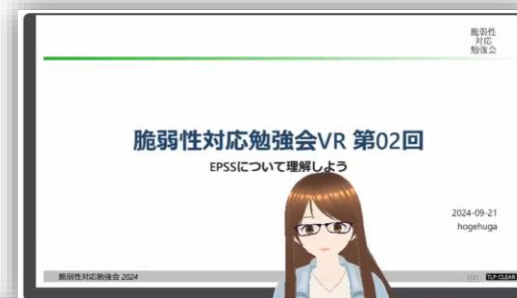
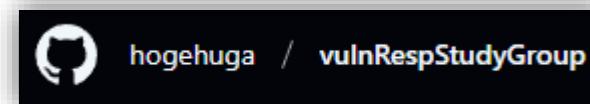
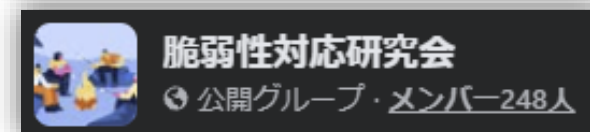
間違いや最新の情報は、背中を刺すのではなく、共有いただければ幸いです。

脆弱性対応勉強会とは

2019年から開催している、セキュリティに関する勉強会です。

- connpass/Facebook/github で活動しています。
 - connpass: メンバーは1,023人(2025-02-11現在)で、勉強会の告知に利用
 - <https://zeijyakuseitaioukenkyukai.connpass.com/>
 - Facebook: メンバーは248人(2025-02-11現在)で、イベント告知と相談的に利用
 - <https://www.facebook.com/groups/zeijyakuseitaioukenkyukai>
 - github: ファイル置き場
 - <https://github.com/hogehuga/vulnRespStudyGroup/>
 - 勉強会としては、28回ほど実施実績があります
- 活動拠点は東京（神田付近？）ですが、県外でも活動をしています。
 - 出張ついでに「出張版 脆弱性対応勉強会」を実施
 - 大阪 2回、名古屋/長崎 1回、札幌 1回（但し誰も現地に来なかった…）
 - VR（アバターを使った録画）も増やします
 - 2回ほど実施

要望があればどこにでも駆けつける勉強会です！



- 2022-04-01 [出張版 脆弱性対応勉強会 #01（札幌）](#)
- 2022-07-08 [出張版 脆弱性対応勉強会 #02（大阪）](#)
- 2023-03-15 [出張版 脆弱性対応勉強会 #03（長崎）](#)
- 2023-06-01 [出張版 脆弱性対応勉強会 #04（大阪）](#)
- 2024-02-09 [出張版 脆弱性対応勉強会 #05（名古屋）](#)

次回予告

2025-02-15（土曜）

- 第14回 脆弱性対応勉強会
（2024年末発表のまとめ）
- InternetWeek/NCA Annual
Conferenceで話した内容をお話しま
す。



2025-03-22（土曜）

- サイバーセキュリティ、失敗と後日談
Vol.01
- みんなの経験を共有しませんか、なLT大
会。



1.脆弱性とは

簡単な、脆弱性というものについての振り返りをします

1. 脆弱性とは
2. 脆弱性の評価
3. 脆弱性の対応判断
4. 脆弱性対応を行う組織の組成
5. まとめ

脆弱性の基本概念

「脆弱性」について、少しまとめます。

- 脆弱性はソフトウェアのバグです。
 - 検出方法はさまざまです。
- WEBサイトは、WEB脆弱性診断が必要
 - プラットフォームやソフトウェアは、バージョンや設定ファイルの診断が必要
 - 但し、ベンダがパッケージで公開しているソフトウェアのバージョンはアップストリームと異なる為、バージョン番号でのそのままの比較はできない
- ソフトウェア間で依存関係関係があり、他のソフトウェアの影響を受ける場合があります
 - 例えば、Apache httpdは、HTTPS通信を行うためにOpenSSLのライブラリを使用しているため、OpenSSLに脆弱性があればApache httpdにも影響がある（Heartbleed CVE-2014-0160）

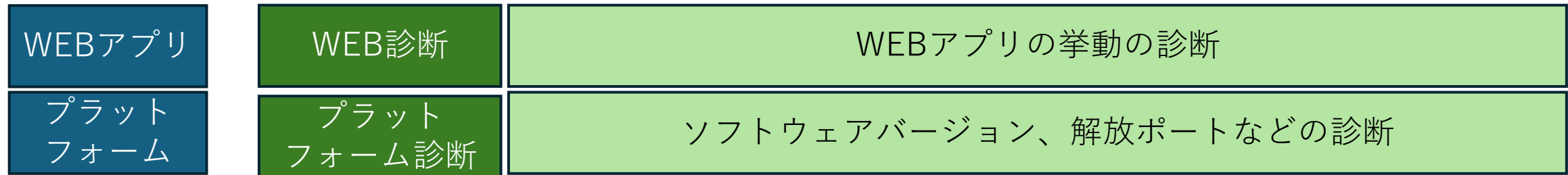
ソフトウェアのバグ自体は無くすることが難しく、早く発見して対応するというアプローチになります。

脆弱性の基本概念：補足

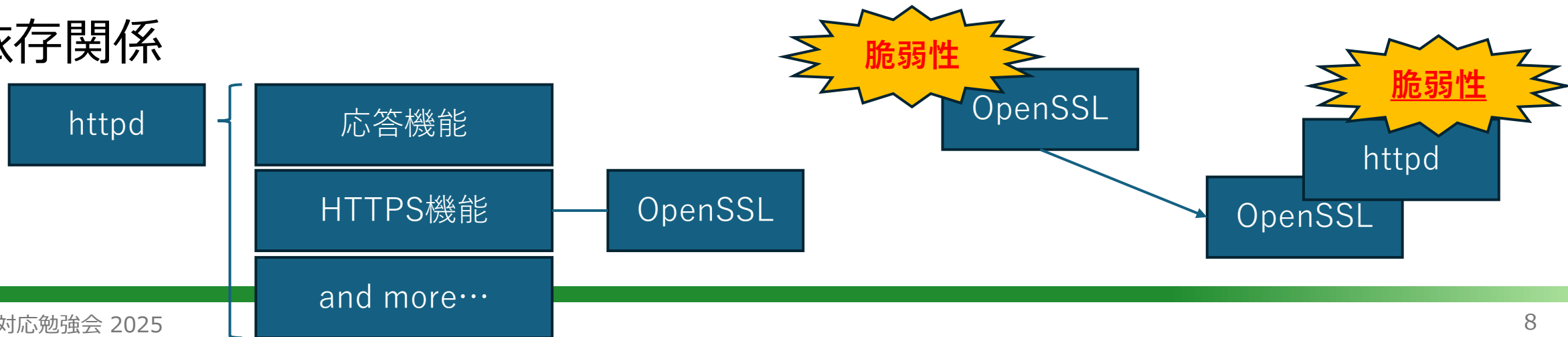
脆弱性



診断



依存関係



2.脆弱性の評価

脆弱性について、どのように評価をするかを例示します

1. 脆弱性とは
2. 脆弱性の評価
3. 脆弱性の対応判断
4. 脆弱性対応を行う組織の組成
5. まとめ

脆弱性と評価

発見された脆弱性はすべて解決するのが理想ですが、現実にはそのようにはできません。

- システムの数により脆弱性の総数が多い、対応工数が非常にかかる、etc
- 全てに対して対応ができない以上、影響があるものを選別して対応する必要がある
 - 事業への影響（=リスク）を評価し、影響度が高いものから対応する、等

リスクを評価する際には、脆弱性それ自体と、おかれている環境に分けて考える必要があります。

この章では、“脆弱性それ自体”の評価についてのフレームワークについて紹介します。

- SCAP
- EPSS
- KEV
- 評価が難しいもの

脆弱性と評価：SCAP (Security Content Automation Protocol)

脆弱性評価でよく使われるCVSS Base Scoreは、SCAPという枠組みで提供されている情報です。

SCAPとは、米国のFISMA(連邦情報セキュリティマネジメント法: Federal Information Security Management Act)に対応すべく、情報セキュリティ対策の自動化と標準化を目指して開発されました。

<https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol/>

SCAPに内包されているものと、よく利用されているものの概要を示していきます。

- CVE
- CVSS(v3, v4), BaseScore/Vector
- CPE
- OVAL
- Scanner: oscap



脆弱性と評価 : SCAP : Specifications (v1.3)

Specifications

- Protocol
 - SCAP
- Tools
 - SCAP Content Validation Tool
- Languages
 - XCCDF : The Extensible Configuration Checklist Description Format
 - OVAL® : Open Vulnerability and Assessment Language
 - OCIL : Open Checklist Interactive Language
 - Asset Identification
 - ARF : Asset Reporting Format
- Identification schemes
 - CCE™ : Common Configuration Enumeration
 - CPE™ : Common Platform Enumeration
 - Software Identification (SWID) Tags
 - CVE® : Common Vulnerabilities and Exposures
- Metrics
 - CVSS : Common Vulnerability Scoring System
 - CCSS : Common Configuration Scoring System
- Integrity
 - TMSAD : Trust Model for Security Automation Data

脆弱性対応/管理をしている場合、SCAPの一部の情報を利用していると思います。

これらについて概説を行います。

- CVE
- CVSS
- CPE
- OVAL

脆弱性と評価：SCAP：CVE

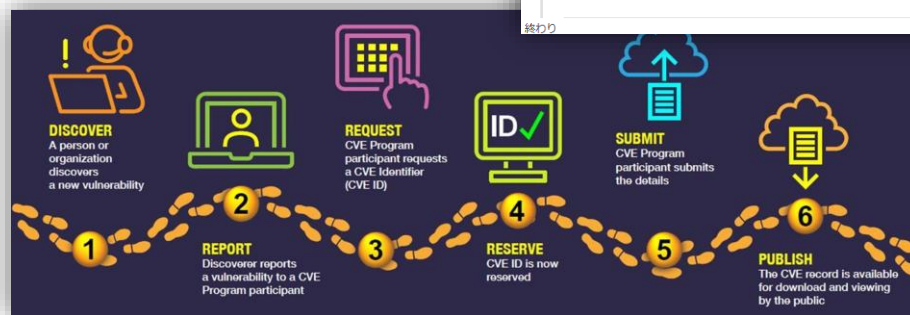
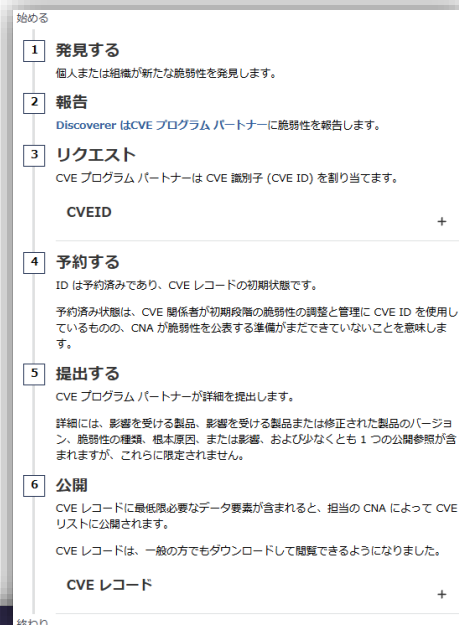
脆弱性を一位に識別するための識別子です。

- CVE-IDが割り当てられます。

➤例) CVE-2024-21762

- 登録プロセス

- Discover
- Report
- Request
- Reserve
- Submit
- Publish



登録プロセス

<https://www.cve.org/About/Process>

CVE[®] プログラムの使命

公開されているサイバーセキュリティの脆弱性を特定、定義、カタログ化します。

現在、[ダウンロード](#)または[検索](#)🔍

Required CVE Record Information

CNA: Fortinet, Inc.

Published: 2024-02-09 Updated: 2024-02-09

Description

A out-of-bounds write in Fortinet FortiOS versions 7.4.0 through 7.4.2, 7.2.0 through 7.2.6, 7.0.0 through 7.0.13, 6.4.0 through 6.4.14, 6.2.0 through 6.2.15, 6.0.0 through 6.0.17, FortiProxy versions 7.4.0 through 7.4.2, 7.2.0 through 7.2.8, 7.0.0 through 7.0.14, 2.0.0 through 2.0.13, 1.2.0 through 1.2.13, 1.1.0 through 1.1.6, 1.0.0 through 1.0.7 allows attacker to execute unauthorized code or commands via specifically crafted requests

CWE 1 Total

[Learn more](#)

- CWE-787: Execute unauthorized code or commands

CVSS 1 Total

[Learn more](#)

Score	Severity	Version	Vector String
9.6	CRITICAL	3.1	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:H/RL:W/RC:C

Product Status

[Learn more](#)

Vendor	Product
Fortinet	FortiProxy

Versions 7 Total

Default Status: unaffected

Affected

- affected from 7.4.0 through 7.4.2
- affected from 7.2.0 through 7.2.8
- affected from 7.0.0 through 7.0.14
- affected from 2.0.0 through 2.0.13
- affected from 1.2.0 through 1.2.13

CVE.orgでの検索

<https://www.cve.org/CVERecord?id=CVE-2024-21762>

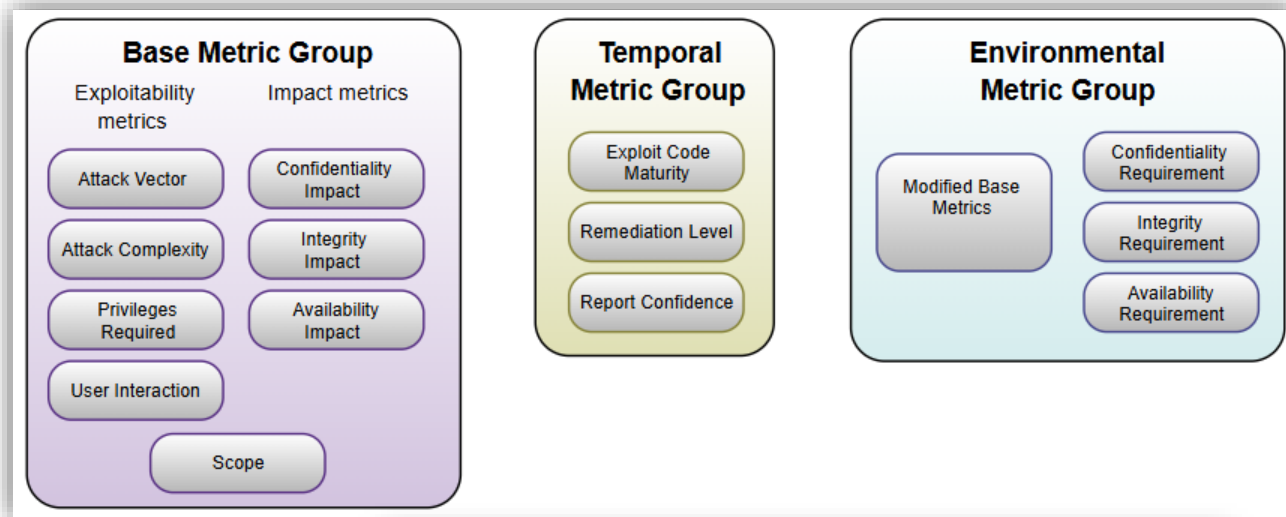
脆弱性と評価：SCAP：CVSS

脆弱性を評価するフレームワークです。

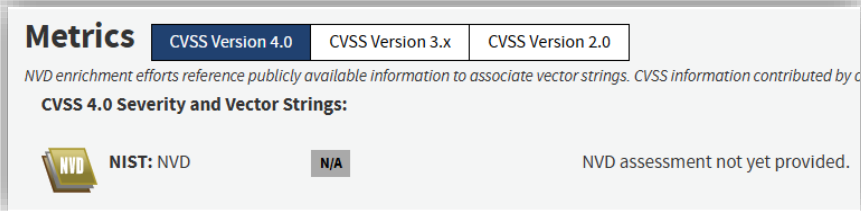
- CVSS Base Scoreとして親しまれているものを含みます。
- 2023年11月01日にv4がリリースされていますが、実用上はv3がまだ使われています。
- 本項目では**v3の説明**をします。基本はv4でも同じです。

脆弱性に対し以下の情報を定義します。

- Base Metric Group
- Temporal Metric Group
- Environmental Metric Group



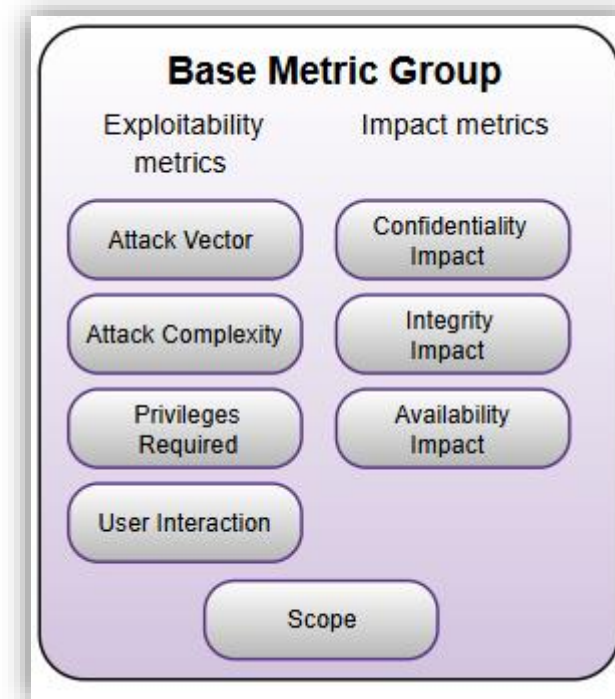
一般的に必要なBaseMetricGroupの説明をします。



脆弱性と評価：SCAP：CVSS：Base Metrics

脆弱性それ自体の評価です。

- Exploitability metrics
 - Attack Vector : どこから攻撃できるか (N/A/L/P)
 - Attack Complexity : 攻撃の複雑さ (L/H)
 - Privileges Required : 必要な権限 (N/L/H)
 - User Interaction : ユーザ関与 (N/R)
- Scope : 影響は対象スコープを超えるか (U/C)
- Impact metrics
 - Confidentiality Impact : 機密性への影響 (H/L/N)
 - Integrity Impact : 完全性への影響 (H/L/N)
 - Availability Impact : 可用性への影響 (H/L/N)



以下のように表現します。

- CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

脆弱性と評価 : SCAP : CVSS : Base Score

CVSS Base Scoreは、Base Metricsを基に算出されています。

もし、Temporal/Environmental Metricsがあれば、この数字から軽減されて計算されます。

- しかしながら、時間的要因や環境的要因はBase Scoreから切り離して言利用することが多いので、あまり使いません。

Severityはスコアから決定されます。

- None : 0.0
- Low : 0.1-3.9
- Medium : 4.0-6.9
- High : 7.0-8.9
- Critical : 9.0-10.0

Base Score: 9.8 CRITICAL

Base Score

9.8 (Critical)

Attack Vector (AV): Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC): Low (L) High (H)

Privileges Required (PR): None (N) Low (L) High (H)

User Interaction (UI): None (N) Required (R)

Scope (S): Unchanged (U) Changed (C)

Confidentiality (C): None (N) Low (L) High (H)

Integrity (I): None (N) Low (L) High (H)

Availability (A): None (N) Low (L) High (H)

CVSS calculator3.1

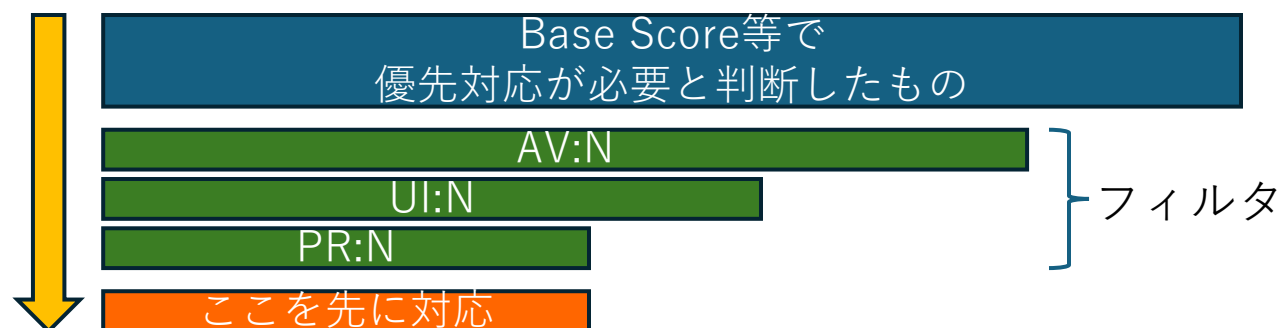
<https://www.first.org/cvss/calculator/3.1>

脆弱性と評価：SCAP：CVSS：BaseScore：補足

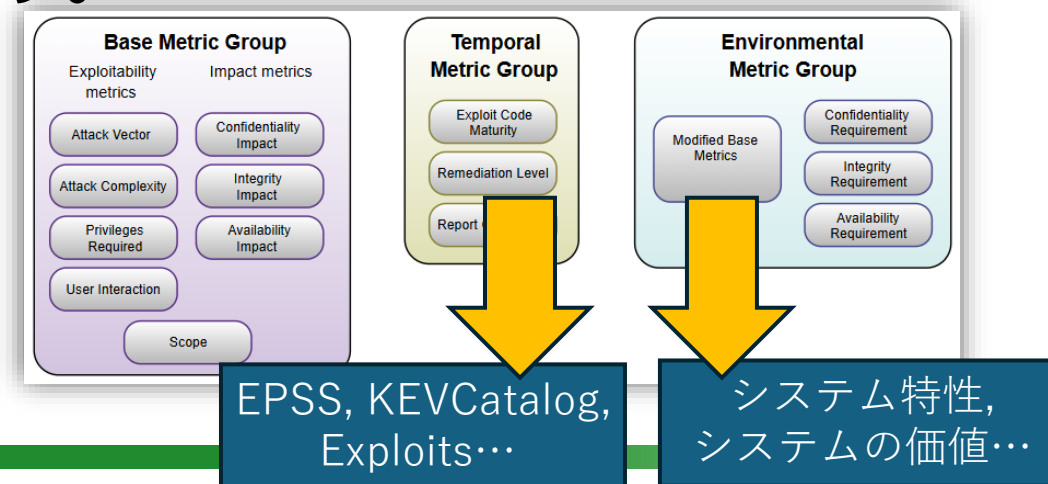
Temporal Metric Group/Environmental Metric Groupは、あまり使われないことが多いし、他の方法で代用が可能なので、CVSSで無理に使うことはないと思われます。

Base Metric Groupは、攻撃者の攻撃何度に影響するものであるため、Base Scoreとともに利用されることが多いです。

BaseScoreが一定以上 AND AV:N AND (UI:N OR PR:N)を優先する などの判断で対応優先度を絞り込むことがあります。



※尚、現実にはこんなに減りません。誇大イメージです



CPEを見てみると発見があるかも



cpe:2.3:a:isamu_kaneko:winny:2.0b5.7:*:*:*:*:*:*

公開日: 2010/08/20 最終更新日: 2010/08/20

JVN#21471805

Winny におけるバッファオーバーフローの脆弱性

緊急

概要

Winny には、バッファオーバーフローの脆弱性が存在します。

影響を受けるシステム

- Winny 2.0b7.1 およびそれ以前

詳細情報

Winny には、バッファオーバーフローの脆弱性が存在します。

なお、本脆弱性は JVN#91740962 および JVN#74294680 と

想定される影響

遠隔の第三者によって、任意のコードを実行される可能性があります。

対策方法

Winny を使用しない

Winny の使用を停止してください。

CVE-2016-4853 Detail

MODIFIED

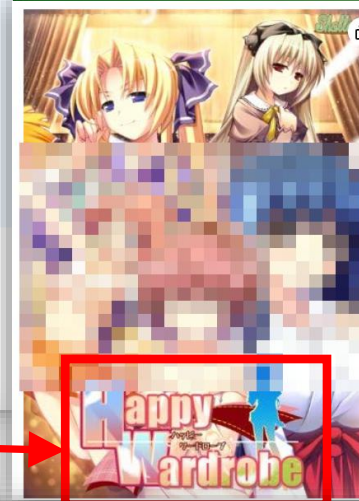
This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Description

AKABEI SOFT2 games allow remote attackers to execute arbitrary OS commands via crafted saved data, as demonstrated by Happy Wardrobe.

cpe:2.3:a:akabei_soft2:happy_wardrobe:*:*:*:*:*:*

[Show Matching CPE\(s\)](#)



Happy Wardrobe
初回版 [アダルト]
ブランド: Shallot
プラットフォーム: Windows
3.0 ★★★★★ 4個の評価
¥1,280 税込
または¥640 月(2か月)。プラン
を選択
prime お届け日時指定便
この商品は、Amazon.co.jp 以外
の出品者(すべての出品を表示)か
ら購入できます。
残り1点 ご注文はお早めに 在庫状
況について
この商品に関する問題を報告
する
アダルト商品につき18歳未満の方は
購入できません。
商品は外から見えないよう厳重に梱
包してお届けします。

対策方法

Winny を使用しない

Winny の使用を停止してください。

ベンダ情報

参考情報

JPCERT/CCからの補足情報

開発者の代理人によると、2010年8月20日時点では刑事事件が係属中のため、開発者による対策の提供予定はありません。

JPCERT/CCによる脆弱性分析結果

謝辞

影響を受けるシステム

有限会社AKABEI SOFT2

- G線上の魔王

上記以外にも影響を受ける製品が多数あるため、詳しくは開発者が提供する情報をご確認ください。

脆弱性と評価：SCAP：CPE（2）

問題点もあります

- 正確性が低い
 - typo等がそこそこある
- バージョン管理があいまい
 - バージョン記載がなかなか難しい



SBOMでは、CPEに代わって purl（Product URL）という新しい方式に移行する流れがある。

- 例えば `pkg:maven/org.apache.commons/commons-lang3@3.12.0`
- 今までより正確にパッケージを特定できるため今後の標準として期待されています。
 - しかしながら現時点では、purlとCVE-IDを直接結びつけるデータベースはないと思われます。

脆弱性と評価 : SCAP : OVAL

OVALはOpen Vulnerability and Assessment Languageで、OSやソフトウェアの脆弱性情報を標準化して表現するXMLフォーマットです。

- 通常、脆弱性スキャナなどが内部で利用し、ユーザが直接利用することはありません。
- OSベンダーが提供する脆弱性情報を含みます（Red Hat, Debianなど）
- OSの特定のバージョンが脆弱性を持つかを判定する、ものです。

OSベンダーが出しているため、バックポートに対応しています。

- （今回は時間的に省略）

脆弱性と評価 : SCAP : OpenSCAP

OpenSCAP (oscapコマンド) はSCAPの仕組みを利用したOSSのツールです。

- 主にLinux環境で動作し、Red Hatの方がメンテナンスしているようです
- このコマンドを利用し、以下のことができます。



➤脆弱性スキャンや、コンプライアンスチェック (PCI/DSSやCIS Benchmark等) が可能

```
ex) # oscap oval eval --report /var/www/html/vulnerability.html rhel-8.oval.xml
```

- Scap Security Guide (SSG) から、チェック用コンテンツを取得できます
- Windowsでは、SCAP Workbench (GUI版といえそう) を使う必要がありそうです。
 - oscapのようなコマンドはなさそうです。数年前にクロスコンパイルだったので試しましたが、ビルドできませんでした。

日本では (も ?) あまり使われていないようですが、PCI/DSSやCIS Benchamrkに対応する場合は有効かもしれません。

脆弱性と評価：KEV Catalog (Known Exploited Vulnerabilities Catalog)

「既知の悪用された脆弱性カタログ」と呼ばれるもので、既に悪用が確認されており、対処方法があるものがリストとして記載されています。

- <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- 大統領令 拘束力のある運用指令22-1 (binding opera.. BOD 22-01) により、連邦政府機関に於いては対応が義務付けられています。
- 公表されて2週間で適用することが求められる (例外あり)



米国内で悪用が確認されたのであり、他国で同様に悪用されているかとは言えませんが、攻撃はグローバルになされていることを考慮すると、日本でも参考にしてよいと思われます。

最低限、ここに記載のある脆弱性是对処すべきと考えます。


脆弱性と評価：KEV Catalog：補足

このサイト、非常に見ずらい気がするので、CSV/JSON辺りを落としてみるのが良いと思います。

以下の項目があります

- cveID
- vendorProduct、product
- vulnerabilityName
- dateAdded、dueDate
- shortDescription
- requiredAction
- knownRansomwareCampaignUse
- notes
- cwes

TRIMBLE | CITYWORKS

 [CVE-2025-0994](#)

Trimble Cityworks Deserialization Vulnerability: Trimble Cityworks contains a deserialization vulnerability. This could allow an authenticated user to perform a remote code execution attack against a customer's Microsoft Internet Information Services (IIS) web server.

Related CWE: [CWE-502](#)

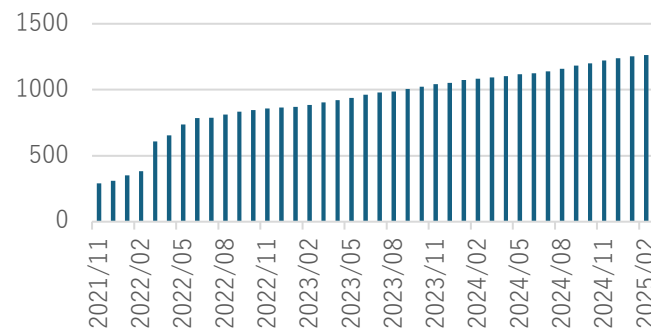
Known To Be Used in Ransomware Campaigns? **Unknown**

Action: Apply mitigations per vendor instructions or discontinue use of the product if mitigations are unavailable.

Date Added: 2025-02-07
Due Date: 2025-02-28

[Additional Notes +](#)

KevCatalog登録総数



```
{
  "cveID": "CVE-2025-0994",
  "vendorProject": "Trimble",
  "product": "Cityworks",
  "vulnerabilityName": "Trimble Cityworks De
  "dateAdded": "2025-02-07",
  "shortDescription": "Trimble Cityworks con
  Information Services (IIS) web server.",
  "requiredAction": "Apply mitigations per v
  "dueDate": "2025-02-28",
  "knownRansomwareCampaignUse": "Unknown",
  "notes": "https://learn.assetlifecycle.tri
  https://nvd.nist.gov/vuln/detail/CVE-2025-
  "cwes": [
    "CWE-502"
  ]
}
```

脆弱性と評価：EPSS（Exploit Prediction Scoring System）

FIRST管理下の(あまり日本語訳しませんが)悪用予測スコアリングシステムで、今後30日以内に脆弱性が悪用される確率を0-1（0%-100%）示すものです。

- CVEデータやExploit情報、脅威インテリジェンスや機械学習等を用いて、可能性を予測しています。



データの読み方は、いろいろできるので、議論があります。

- 確率の数値として、どのように取り扱うかの議論の余地があります。
- EPSSは確率だけを示しているので、CVSS Base Scoreなどと併せて利用する必要がある

個人的には絶対値ではなく、相対比較で使われるのが望ましいように思われる

脆弱性と評価：EPSS：補足

- EPSSは、先述のKEV Catalogも参照しています。
- データは、APIで個別に取得するか、.csv.gzで取得するかの2択になります。
- 以下の内容が含まれます
 - cve
 - epss
 - percentile
 - date
- 弊研究会のツールも使えます
 - hogehuga/epss-db
 - ✓ EPSSデータを時系列で保管し、SQL文で分析できます
 - ✓ Splunk/Elasticを使え？ そんなリソースは、無え！（mysql）
 - hogehuga/threatWatchDog
 - ✓ 前日比でepssが高くなったものを表示します

```
{
  "status": "OK",
  "status-code": 200,
  "version": "1.0",
  "access-control-allow-headers": "x-requested-with",
  "access": "public",
  "total": 1,
  "offset": 0,
  "limit": 100,
  "data": [
    {
      "cve": "CVE-2025-0994",
      "epss": "0.055820000",
      "percentile": "0.932400000",
      "date": "2025-02-08"
    }
  ]
}
```

脆弱性と評価：評価が難しいもの

実運用上で脆弱性の評価が難しいものもあります。

- 修正プログラムが存在しない脆弱性

- 問題は判明しているが、対策が無い場合（後述EOL含む）
- Exploitされる前提で事後対応を考える、状況によりサービス停止も視野に入れる

- サポートが終了したソフトウェアの脆弱性

- そもそも、サポートが終了したので脆弱性の検査をせず、更新も提供しない
- 社会的影響を考え、対応される場合もある（ex. 古いWindowsの脆弱性）
- この状況に陥らないように運用やサービス設計をする必要がある
- 対象での修正は難しいため、環境による保護や対応しそうな仮想パッチ製品を使う必要があるが、根本対策ではない為注意は依然必要
- 上記で延命しつつ、リプレイス等を計画する必要がある（が、経営マターかもしれない）

脆弱性と評価：評価が難しいもの：コラム「パッチって言うな（？）」

ソフトウェアのセキュリティ更新プログラムを「パッチ」と呼ぶことがあります、個人的には非常に違和感があります。。

- 旧来であれば、ソースにpatch適用し（`patch` コマンドありましたよね）、コンパイル（make）しなおすことで「修正パッチを適用」していました。

- ``patch -p1 < ../mutt-1.4.2.1i-ja.1/patch-1.4.2.1.tt.ja.1 ; ./configure; make``とか

- 現代では、バイナリで配布されることが多く、プログラム丸ごと/一部を入れ替える形の「更新プログラムの適用」方式になっています。

- セキュリティ更新プログラムとか、更新されたパッケージの適用とかが多い。例外はある。

- パッチワークの意味合いの、あて布をする/継ぎを当てる、だと、まあ該当する…

- そもそも文字数が「パッチ（3文字）」「更新プログラム（7文字）」で短い方が使われる

- patchコマンドじゃないからパッチではない原理教、なだけかもしれない…

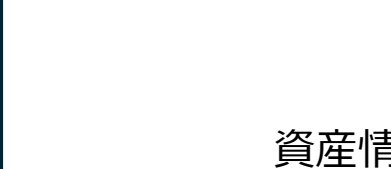
- ✓ 意味が通って更新されれば、呼称は何でもよいのではないかな…

- ✓ patch警察、ではない。。

しかし、この資料でも
パッチ言うてる…

脆弱性を評価/管理するには、持っているソフトウェア等の資産リストが必要になります。

- ソフトウェアの資産をまとめるものとして、SBOMが使われる方向になってきています。

- 公開された脆弱性情報
- 保有資産
- 要対処
- 資産情報で
対象を絞る
- 

- 例えば、ライセンス情報の取り扱いが異なります。

- 脆弱性対応勉強会 2025

3.脆弱性の対応判断

脆弱性について、どのように対応判断を行うかを例示します

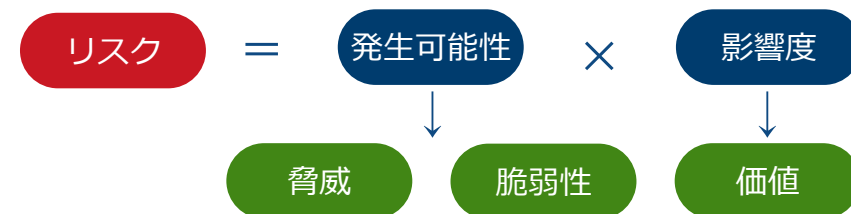
1. 脆弱性とは
2. 脆弱性の評価
3. 脆弱性の対応判断
4. 脆弱性対応を行う組織の組成
5. まとめ

脆弱性対応の判断

脆弱性すべてに対応できれば良いですが、現実的には一度にすべての対応をすることはできません。その為、対硫黄の優先順位をつける「脆弱性トリアージ」を行う必要があります。

これは一般的に、事業へのリスクに注視して優先順位をつけることになり、リスクは以下のようになります。

- リスク = 発生可能性 × 影響度
= 脆弱性 × 脅威 × 価値 (Threat x Vulnerability x Impact)
- 脆弱性 : 脆弱性それ自体の価値。CVSS Base Score等。
 - 脅威 : 脆弱性を利用されるかどうか。システム構成やEPSS等。
 - 価値 (影響) : システムの持つ価値。保有情報や事業継続性への影響等。



リスクが低いと判断したとしても、その脆弱性を無視してよいわけではない点に注意が必要です。

脆弱性対応の判断：脆弱性のトリアージ

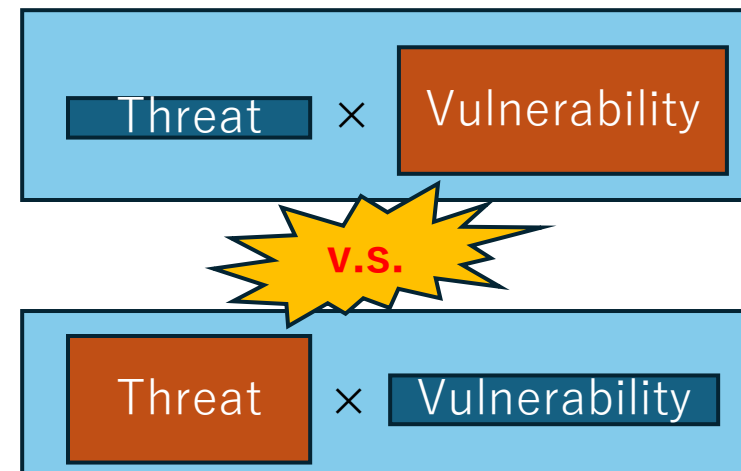
前述のリスクを判断するために、脆弱性トリアージを行います。

一般的には単純化し、脆弱性×機会 で影響が大きいものから対応を行う事になります。

- CVSS Base Scoreが高いものに対し、EPSSが大きい（攻撃確率が高い）ものを優先する、という判断指標はよくあると思います
- しかしながら、個々の脆弱性を一つずつ評価することは、非常に労力がかかります
- その為、ある程度自動化をする必要があります
 - 自動化をしない場合は、ある程度CVSSでめどをつけた上での最終判断程度で利用するのが良いかもしれせん。

脆弱性トリアージの自動化としては、以下が参考になります。

- SSVC
- CVSSとEPSS/KEVCatalog の組み合わせ

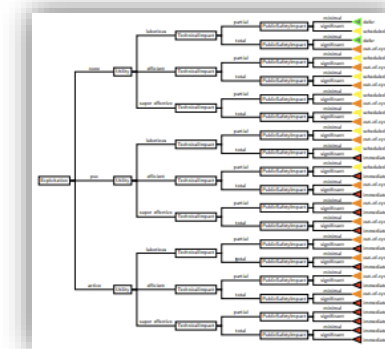


脆弱性のトリージ：SSVC

SSVCは、Stakeholder-Specific Vulnerability Categorizationで、脆弱性対応の優先度を定めるための意思決定フレームワークです。

従来のCVSSとの比較すると以下ようになります。

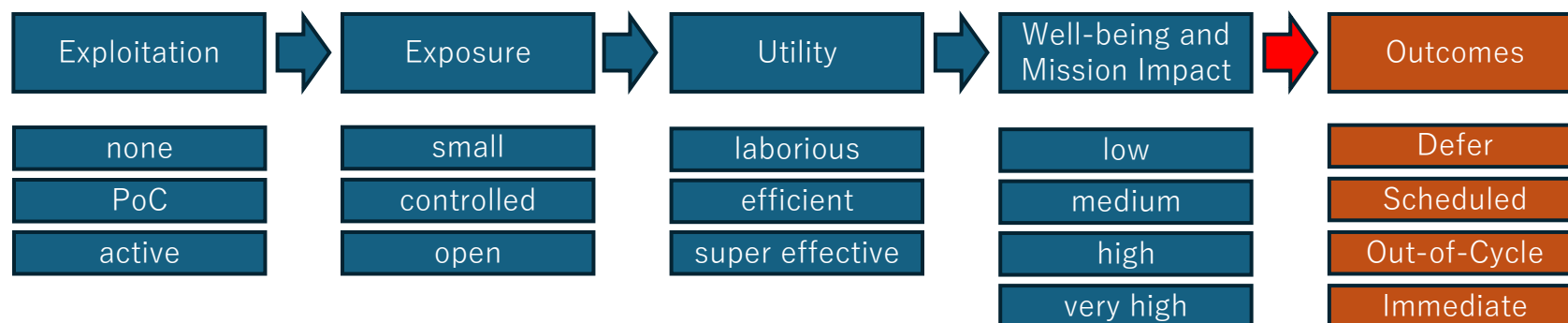
- CVSSはスコアのみで判断しがちである
- SSVCは組織の役割ごとに異なる優先度を考慮する



決定木（Decision Tree）を利用して脆弱性を分類します。決定木は複数用意されており、日本ではユーザ企業がDeployer Treeを使う場合が多いようです。

優先度として以下が出力されます。

- Defer
- Scheduled
- Out-of-Cycle
- Immediate



弱性のトリアージ：SSVC：例示

実際のCVE-IDを当てはめて確認してみます。（CISA Coordinator Treeを使います）

- CVE-2024-21762

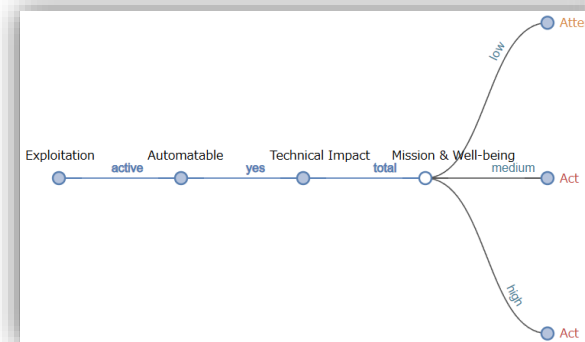
- Vulnrichmentによりると、以下のような状況です。

- Exploitation: **active** / Automatable: **yes** / Technical Impact: **total**

- CISA Coordinator Tree(v1)を見ると、Mission&Well-beingは手動入力が必要

- Mission Prevalence : (企業等の) 使命/missionにどの程度不可欠か
- Public Well-being Impact : 公衆衛生（身体/環境/経済/心理 的）への影響
 - 対象システムと環境によるので、自身で定義する必要がある。
 - lowの場合はAttend、そのほかはActで対応が必要となる

```
{  
  "metrics": [  
    {  
      "other": {  
        "type": "ssvc",  
        "content": {  
          "id": "CVE-2024-21762",  
          "role": "CISA Coordinator",  
          "options": [  
            {  
              "Exploitation": "active"  
            },  
            {  
              "Automatable": "yes"  
            },  
            {  
              "Technical Impact": "total"  
            }  
          ]  
        }  
      }  
    ],  
  ]  
}
```



注意点

- 決定木に任せる、という決断ができるか

- CISA Coordinator Treeなら、Vulnrichment/NVDでDecision pointの値が公開されています。
- NVDお勧めで決定できるので、CISA Coordinator Treeを使ったほうがいいかも

- そもそも、SSVC自動判定な「OSSツール」ってあったっけ…

- ウチで作ろうか…。というか自動判定しないとこんなの使えんよ。

脆弱性のトリアージ：組み合わせ

CVSS Base ScoreやEPSS, KEV Catalogなどを組合わせて、ある程度自動で判断を行うツールが存在します。

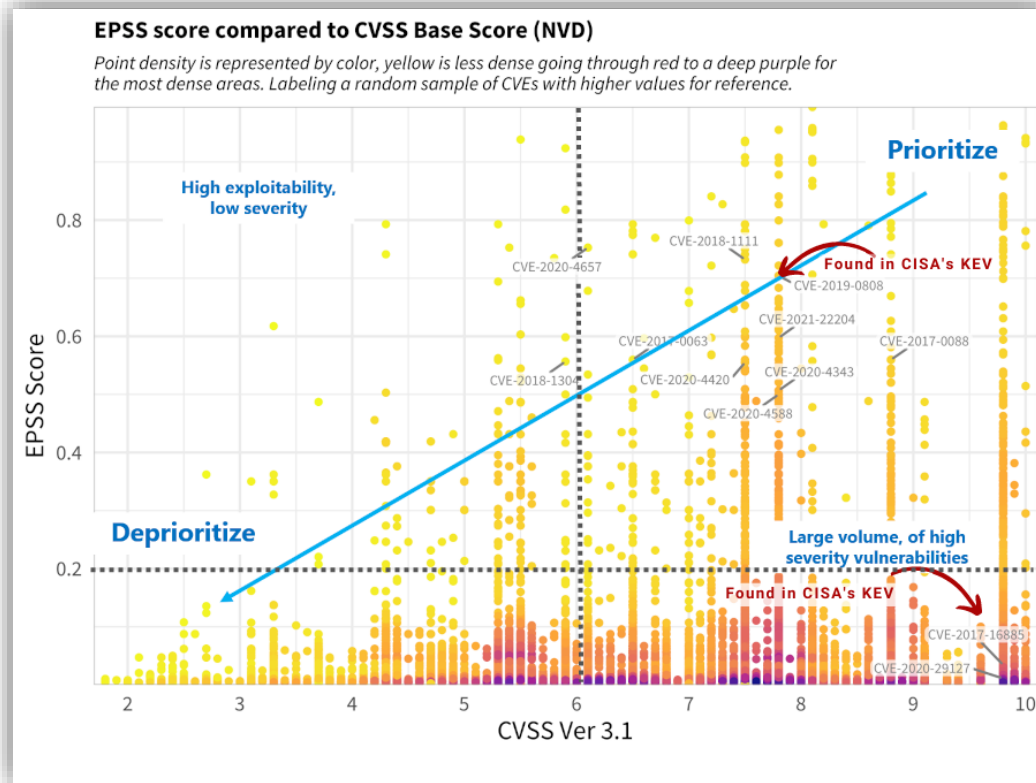
- CVE_Prioritizer
- SploitScan

考え方としては非常に参考になる為、ここでは各ツールを紹介します。

脆弱性のトリアージ：組み合わせ：CVE_Prioritizer

https://github.com/TURROKS/CVE_Prioritizer

- CVSS, EPSS, KEV Catalog, VulnCheckのコミュニティリソース(NVD++, KEV)を組み合わせ、脆弱性のパッチ適用優先順位付けを支援するツール



優先度	説明
Priority 1+	KEV Catalogに登録がある
Priority 1	右上象限のCVE
Priority 2	右下象限のCVE
Priority 3	左上象限のCVE
Priority 4	左下象限のCVE

脆弱性のトリアージ：組み合わせ：SploitScan

<https://github.com/xaitax/SploitScan>

- CVE_Prioritizer同様、CVE, EPSS, KEV Catalogなどを使用します
 - 比較すると、脆弱性スキャナ連携機能やAI利用など、高機能

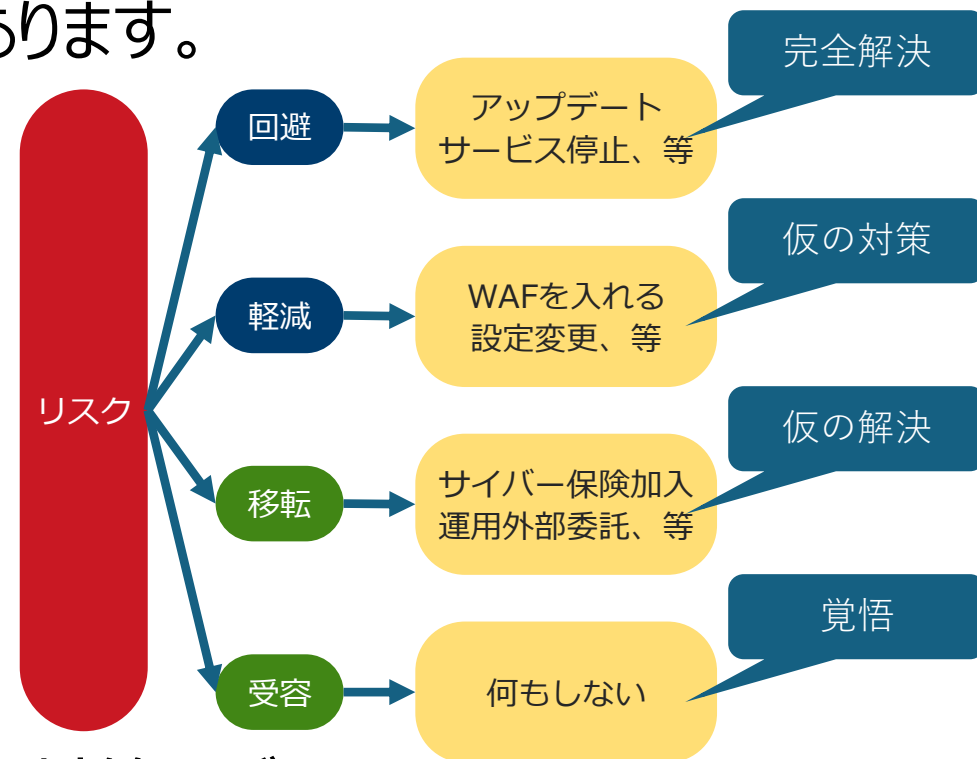


優先度	CVSS Score	
A+	KEV Catalogにある または Exploitがある	パッチ適用のリスクと緊急性が最も高い
A	CVSS \geq 6.0 かつ EPSS \geq 0.2	深刻度が高く、悪用される可能性がかなり高い
B	CVSS \geq 6.0 だが EPSS $<$ 0.2	重大度は高いが、悪用される可能性は低い
C	CVSS $<$ 6.0 かつ EPSS \geq 0.2	重大度は低いが、悪用される可能性は高い
D	CVSS $<$ 6.0 かつ EPSS $<$ 0.2	重大度が低く、悪用される可能性が低い

脆弱性のトリアージ：リスクと対応

リスクにどのように対応するか、対応を決める必要があります。
おおよそ以下のような対策を取ることが求められます。

- 回避：原因を根本から取り除く
- 軽減：影響を軽減する
- 転移：リスクを他に転嫁する
- 受容：リスクを受け入れる



リスク移転としてサイバー保険を記載していますが、これは発生を抑止できるものではなく、発生時の被害対策なだけであり、対策とは言えません。

また、需要に関しても、発生するリスクに対策コストが見合わない場合の想定です。

脆弱性のトリアージ：対応速度

脆弱性にどの程度の速度で対応を行うかも決める必要があります。

SSVCでの定義などを見るに、定期メンテナンスのタイミングを基に、おおよそ以下の5段階を定めるのが主流になっていると思われます。

- 今すぐ/緊急
- 次のメンテナンスタイミングより早く
- 次のメンテナンスタイミング
- 四半期のメンテナンスタイミング
- 注視（パッシブな対応）

対応速度

緊急：今すぐ

次のメンテナンスタイミング
より、早く

次のメンテナンスタイミング

四半期のメンテナンスタイミング

注視：様子見：対応除外

脆弱性のトリアージ：実際の評価例

トリアージの評価例として、Heartbleedでの悪用事例を挙げます。

- CVSS Base Scoreは10.0で、数値だけ見ると危険なものでした。
- 初期の段階では本当に悪用されるのかの議論がありました。
- 結果的に広範囲のシステムが影響を受け、企業は迅速な対応を迫られましたが、実際に悪用事例が確認されるまでには時間がかかりました。



このようにCVSS BaseScoreだけでは優先度を判断できず、KEV CatalogやEPSSなどと組み合わせて評価することが有効なことがあります。

Heartbleed：OpenSSLの脆弱性で、TLSのHeartbeat拡張機能にバグがあり、サーバのメモリが読み取れてしまう。

また優先度判断として、すぐに適用するリスクも考慮する必要がある

- パッチ適用に一時的なサービスダウンが必要
- 証明書再発行が必要なため、対応コストが高い
- 影響度は大きいが直ちにパッチ適用するのが正解なのかという議論もあった

これらの事業判断踏まえて、評価を行う必要があります。（後からなら、なんとでも言えるが…）

脆弱性のトライージ：どのように始めるか、使うか

個人的には、どこまで自動化を信じられるか、次第と思っています。

- トライージフレームワークに全てお任せしてよい。必要なら自分で評価確認して変えるよ。
 - SSVC辺りがおすすめ。Misson&Well-beingをシステムなどの範囲で決め打ちすると自動化ができる。その際は、CISA Coordinator Treeを使うと、その他のDecision Pointを自動で決定できる。
 - Outcomeを見て、必要に応じて優先順位を変える
- 誰かが決めた評価は信じられない！今のCVSS判断から逃れられない！
 - CVSSとその他の情報で頑張りましょう。
 - ✓ まずは先に、トライージポリシーを決めるところから。CVSS BaseScore8.0以上、では無理よ？
 - ✓ EPSS, KEV Catalog, Vulnrichment, Exploit情報などを活用する
 - 評価方法が分かるツールを使う
 - ✓ 残存CVE-IDを脆弱性スキャナで集め、CVE_PrioritizerやSploitScanにぶち込む
 - お金を払って、良いと思われる脆弱性管理ツールを買う

トライージの手間と、それによって得られる効果（工数削減/選別による速度向上など）を考慮して、手段は考えたほうが良いです。まずは、トライージポリシーを作ることが優先ではあります。

4.脆弱性対応を行う組織の組成

脆弱性対応を行う場合、どのような機能が必要なのかを例示します

1. 脆弱性とは
2. 脆弱性の評価
3. 脆弱性の対応判断
4. 脆弱性対応を行う組織の組成
5. まとめ

セキュリティ対応組織、という観点からの脆弱性管理

ここからは、脆弱性管理というシステム側からの視点ではなく、組織全体のセキュリティ対応等言う観点から見てみます。

組織内のセキュリティ対応を行う組織（CDC: Cyber Defence Centre）についてのフレームワークとして、ITU-T X.1060というものがあります。

X.1060では、CDCとして求められる機能がまとめられています。

今回は、CDCの機能と脆弱性管理で必要な機能について比較検討してみます。

ITU-T X.1060の概要

ITU-X 1060とは、簡潔に以下のようなものです。

- <https://www.itu.int/rec/T-REC-X.1060-202106-I>
- 組織のサイバー防衛部門（CDC）のフレームワークを定義したもの
- 組織のセキュリティ運用に必要な機能を体系的に整理されている

日本語版として以下があります。

- TTC JT-X1060 サイバーディフェンスセンターを構築・運用するためのフレームワーク
 - https://www.ttc.or.jp/document_db/information/view_express_entity/1423
- ISOG-J セキュリティ対応組織の教科書 第3.x版
 - https://isog-j.org/output/2023/Textbook_soc-csirt_v3.html

今回はその中でも脆弱性管理に焦点を当てます。

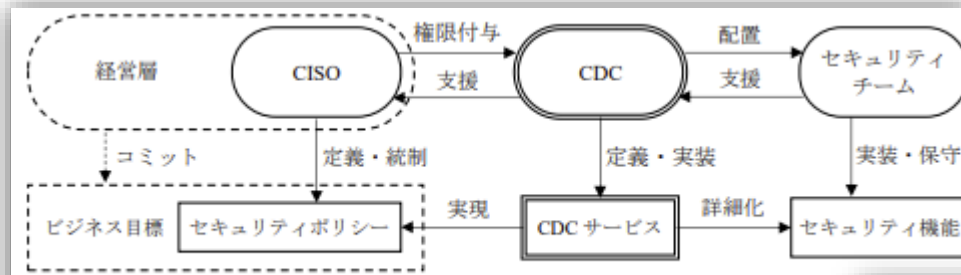


図1 CDCの運営における関係者とその役割

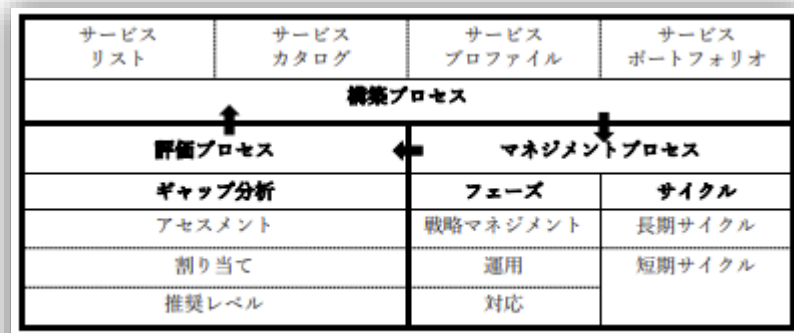


図2 サイバーディフェンスセンターを構築・運用するためのフレームワーク

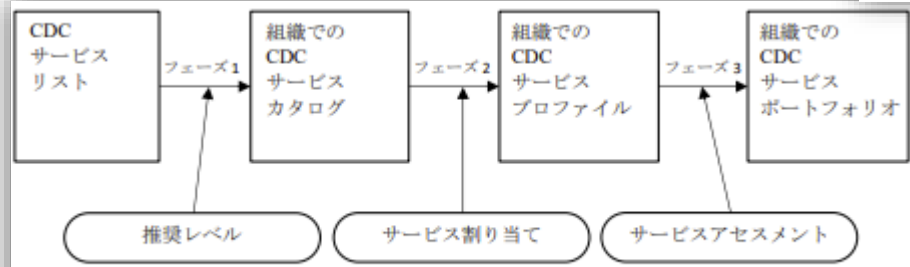


図3 CDCサービスの立ち上げフェーズ

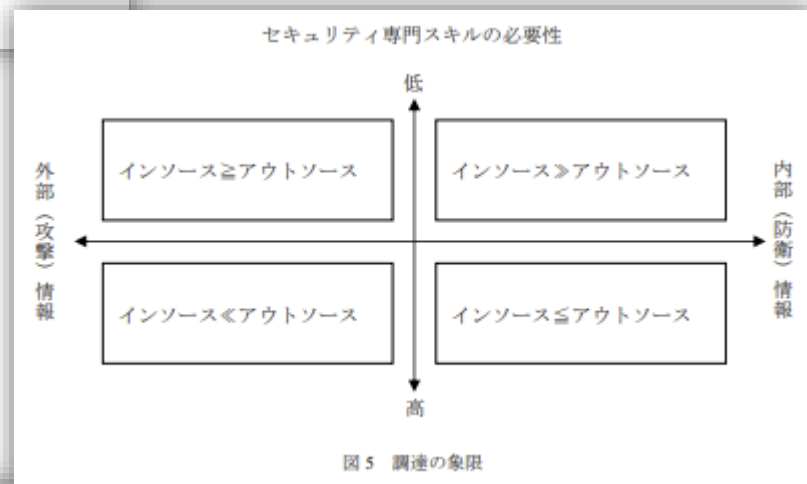


図5 調達の象限

**X.1060は国際標準なので、国ごとに異なるであろう詳細は書かれていない。
日本では、セキュリティ対応組織の教科書がより詳しく記載されているので、そちらを参照されたい。**

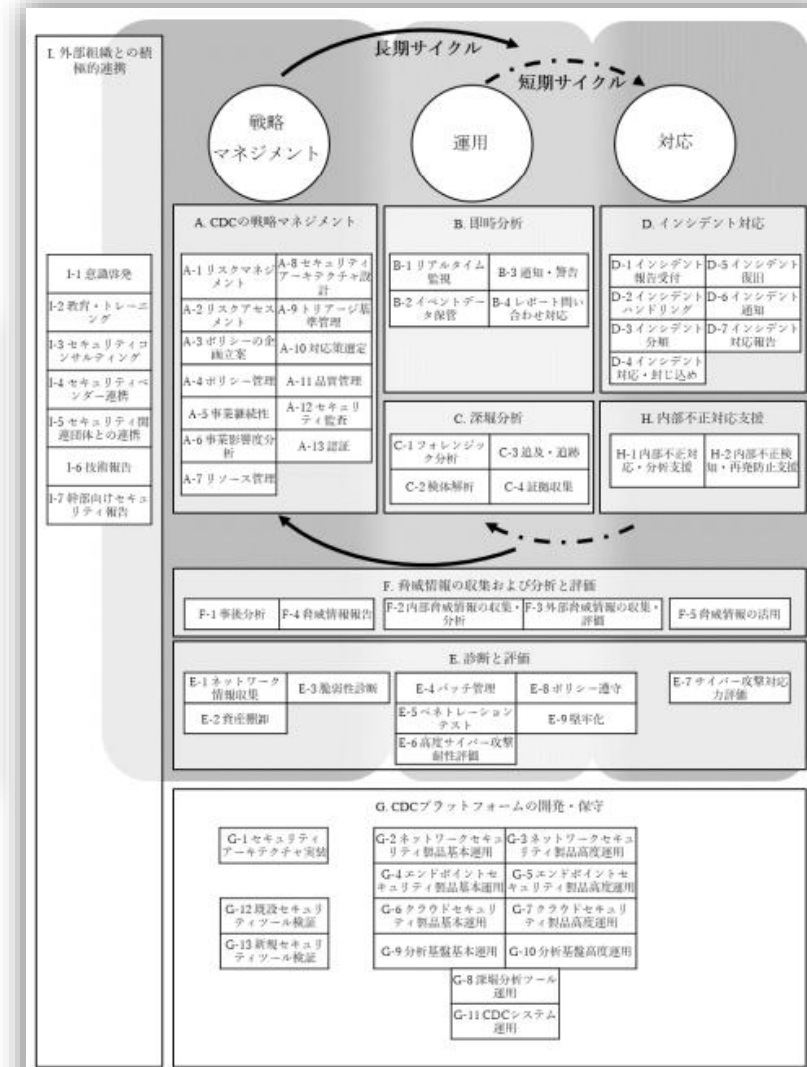
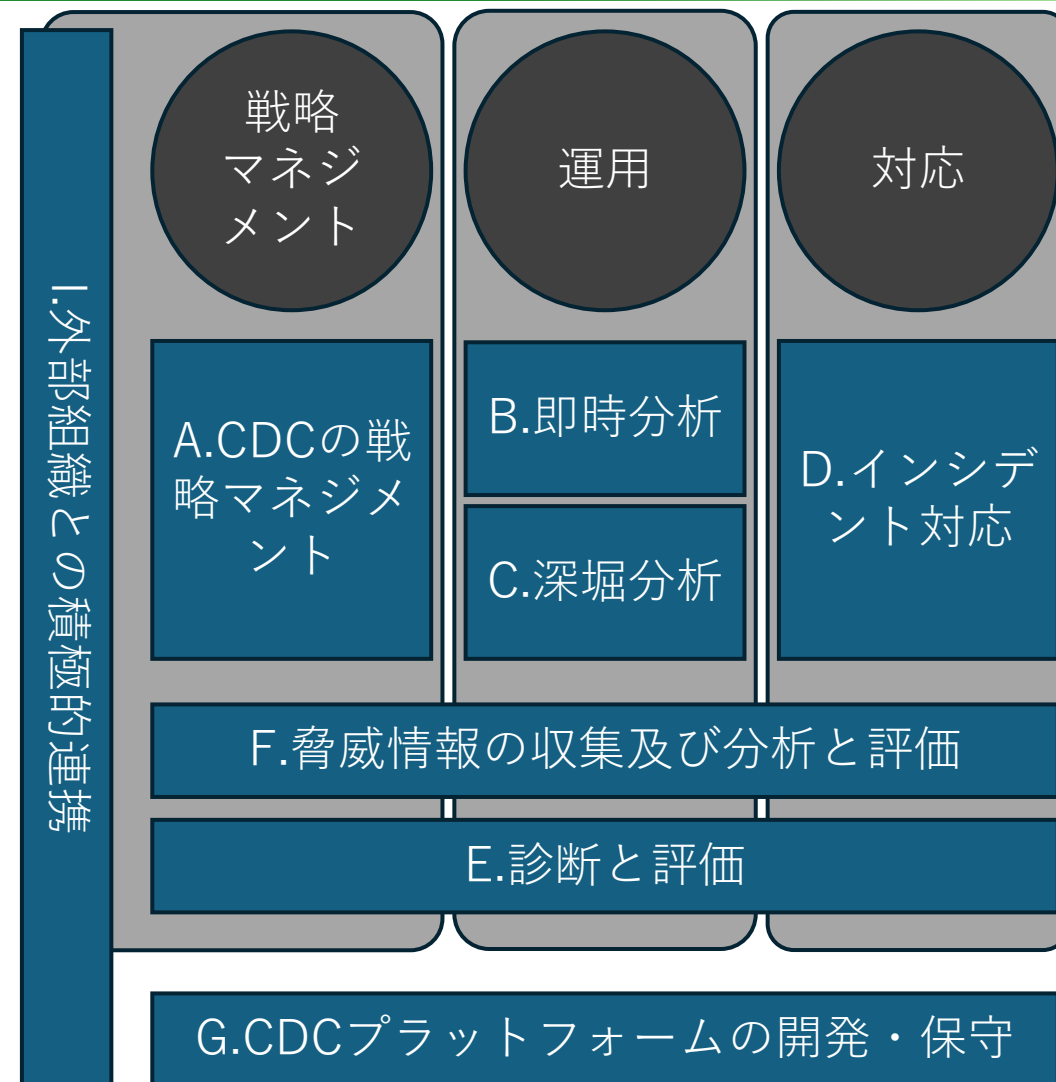


図8 CDCサービスカテゴリー

X.1060の全体像：簡略図

- A) CDCの戦略マネジメント
- B) 即時分析
- C) 深堀分析
- D) インシデント対応
- E) 診断と評価
- F) 脅威情報の収集及び分析と評価
- G) CDCプラットフォームの開発・保守
- H) 内部不正対応支援
- I) 外部組織との積極的連携



ITU-T X.1060 : CDC Service Categoriesの紹介

CDCの機能は多岐に渡りますが、脆弱性管理にはこの部分が必要と考えられます。

これらを重点的に、現状との比較を試みるのが良いと思います。

- A-9 : トリアーシ基準管理
 - 全社のポリシーで合意された範囲内で、トリアーシ基準作成を実現する
- F-3 : 外部脅威情報の収集・評価
 - 新たな脆弱性、攻撃傾向、レピュテーション情報などの外部情報の収集を実現する
- E : 診断と評価
 - 資産棚卸から脆弱性診断、パッチ適用や堅牢化などを実現する
- G-11 : CDCシステム運用
 - 脆弱性管理などの、セキュリティ対応業務に必要なタスクを遂行するシステムの運用を実現する
- I-5 : セキュリティ関連団体との連携
 - 外部コミュニティへの参加を通じ、積極的な情報交換を実現する

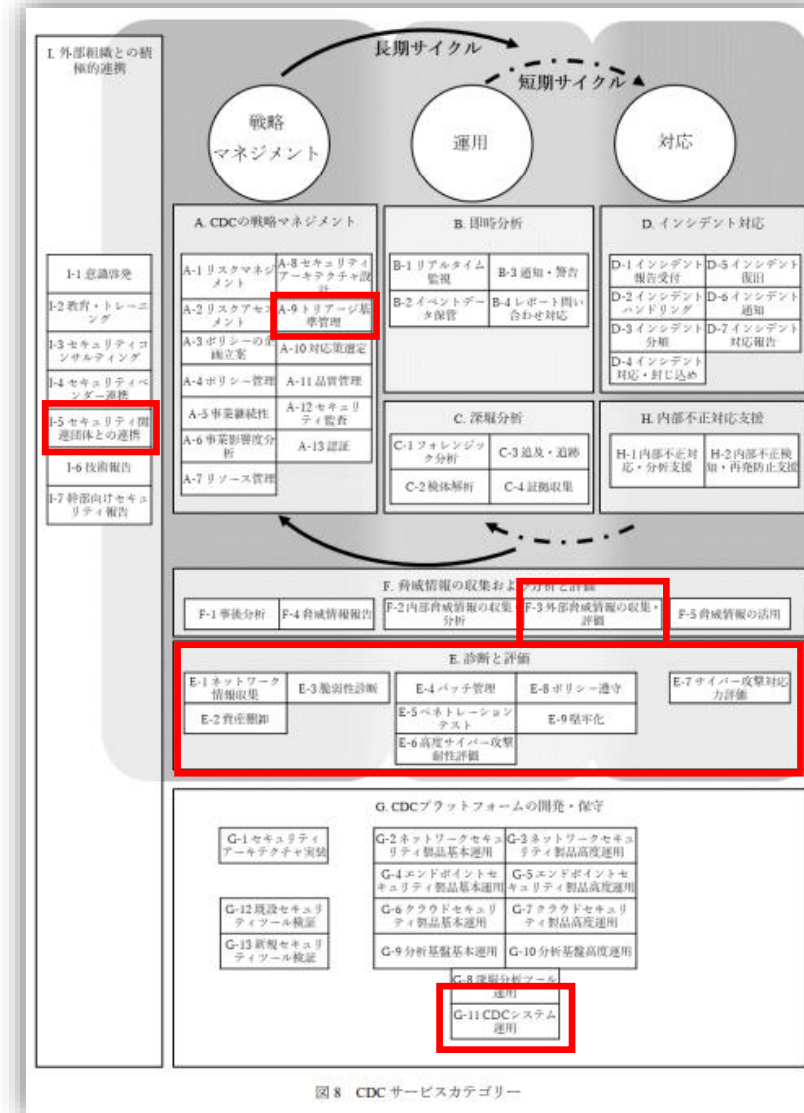


図8 CDC サービスカテゴリ

ITU-T X.1060：脆弱性管理に必要な6つの機能

先ほどの項目について、自組織で不足している/必要としているかを検討する必要があります。

Category	説明	重要な理由
A-9. トリアーージ基準管理	トリアーージ基準の管理	脆弱性の優先順位をどう決めるか？ (KEV Catalog、EPSS、SSVC)
F-3. 外部脅威情報の収集・評価	脅威情報の収集	新しい脆弱性情報を迅速に取得できる仕組みがあるか？ ベンダーの修正情報やパッチリリースを追えているか？
E. 診断と評価	監査・評価	定期的な脆弱性スキャンは実施しているか？ (WEB、プラットフォーム)
G-11. CDCシステム運用	システムの運用保守	脆弱性管理製品は適切に運用できているか？
I-5. セキュリティ関連団体との連携	外部連携	対応ポリシーは、一般的に妥当なのか？ 業界内で協力や相談ができる状況か？

項目すべてを満たすのが目的ではなく、現状ではどこまでの対応が必要かを考慮し手必要な領域を進めることが重要です。

少なくともF-1:Security assessmentができるように、ネットワーク構成や利用ソフトウェア（アセット）の把握はしておく必要があります。これができることがベースになると考えられます。

5.まとめ

なんとなくまとめた風のことを言って、良い感じに終わらせます

1. 脆弱性とは
2. 脆弱性の評価
3. 脆弱性の対応判断
4. 脆弱性対応を行う組織の組成
5. まとめ

まとめ

脆弱性管理のポイント

- 脆弱性は無くならない…→いかに適切に/楽に対応できるようにするかが肝
- 自組織の環境に合わせた評価、それによる優先順位付けが必要
- 組織的な対応フローの決定が不可欠

今後の課題

- SBOMの実効性
- 自動化の更なる発展
- 業界全体での協力と情報共有

今日の内容を自社脆弱性管理にどう生かせるかを考得て頂ければ幸いです。

- 後ほど資料を公開します。

(本音)

- 脆弱性管理、面倒だし、したくないよね…
- プラットフォーム部分は、どうせベンダのセキュリティ更新街なので、最新にすれば責任を押し付けられるのでは？
- 設計段階からContainer/モジュール化を進めて、自動で適用できるようにしたいね
- その為には、テストの自動化で、自動更新と評価とロールバックができる設計が必要ね
- それらができれば、脆弱性/パッチ管理コストが減るので別の物に投資できるし、別の有意義な作業ができるようになるよね

脆弱性管理の話をしているけど、脆弱性管理をしたくないんだ…。
それらが無い世の中になる仕組みを議論したいのよ？

Thank you!

Any Questions?

オープンな議論としてお話ししたい場合は、Facebook/X/LinkedIn/Eight 等でご連絡下さい。

仕事として議論をしたい場合は、kei.inoue@lac.co.jp までご連絡ください。

(どちらもお金が貰えるわけではないので、どちらでもいいです…)

Appendix

参考URL等を記載します

- SCAP
 - IPA:セキュリティ設定共通化手順SCAP概説
 - <https://www.ipa.go.jp/security/vuln/scap/scap.html>
 - 2015/07の資料の為、現状のSCAP 1.3より古いが、概要としては価値がある
 - NIST: Security Content Automation Protocol
 - <https://csrc.nist.gov/projects/security-content-automation-protocol/>
- KEV Catalog
 - CISA: Known Exploited Vulnerabilities Catalog
 - <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- EPSS
 - FIRST: Exploit Prediction Scoring System
 - <https://www.first.org/epss/>
 - hogehuga tools
 - <https://github.com/hogehuga/epss-db>
 - <https://github.com/hogehuga/threatWatchDog>
- Ssvc
 - CISA: Stakeholder-Specific Vulnerability Categorization(SSVC)
 - <https://www.cisa.gov/stakeholder-specific-vulnerability-categorization-ssvc>
 - Carnegie Mellon University: Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization (Version 2.0)
 - <https://insights.sei.cmu.edu/library/prioritizing-vulnerability-response-a-stakeholder-specific-vulnerability-categorization-version-20/>
 - CERT: ssvc-calc (has any Decision Tree 🍌)
 - <https://certcc.github.io/SSVC/ssvc-calc/>
 - CISA: Ssvc Calculator (CISA Coordinator only)
 - <https://www.cisa.gov/ssvc-calculator>

- ITU-T X.1060
 - ITU-T: X.1060 : Framework for the creation of cyber defence centre
 - <https://www.itu.int/rec/T-REC-X.1060-202106-I>
 - TTC: JT-X1060 – サイバーディフェンスセンターを構築・運用するためのフレームワーク
 - https://www.ttc.or.jp/document_db/information/view_express_entity/1423
 - ISOG-J: セキュリティ対応組織の教科書 3.x版
 - https://isog-j.org/output/2023/Textbook_soc-csirt_v3.html
- 過去の発表資料
 - Internet Week 2023
 - <https://www.nic.ad.jp/ja/materials/iw/2023/proceedings/c6/>
 - Internet Week SHOWCASE in 福岡(2024)
 - <https://www.nic.ad.jp/ja/materials/iw/sc-fukuoka/proceedings/c23/c23-Inoue.pdf>