脆弱性情報の上流は安定していない

公開前検査による116日間の全数観測

篠原 俊一・中岡 典弘・神戸 康多

フューチャー株式会社

コンピュータセキュリティシンポジウム 2026

OWS 2F1-2 @ 2026-10-20

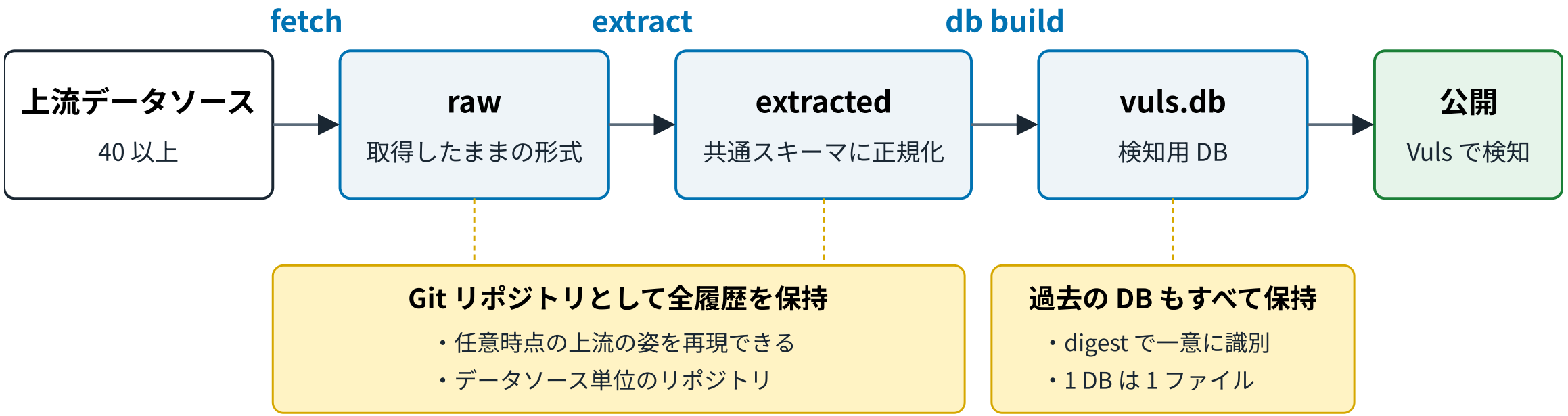
まとめ

1. 脆弱性情報の上流は 2 日に 1 件のペースで **大きく動く**。
   消え、戻り、書き換わる。**大半は正当な変更**。

2. でも、**本当にまずい DB** がたまにできる。
   1 か月分の CVE がまるごと消える。10 万ファイルが半日ごとに入れ替わる。
   自動ビルドを、公開前に **止めないといけない**。

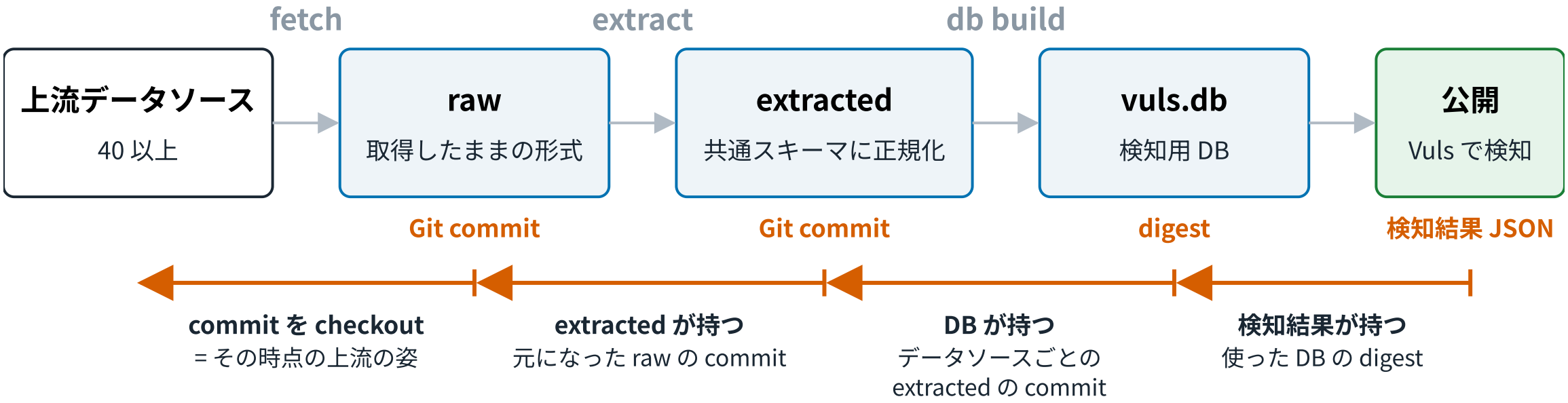
3. 公開前に **前回の DB** との差を見て、**大きな差分** はひとまず止めるようにした。
   116 日間、止めた全件の原因を履歴で特定した。
   本当にまずい DB を 2 回捨てた、少しまずい DB も 1 回捨てた。

基盤: 脆弱性 DB のビルドと公開



- 3 種とも GitHub Container Registry (バイナリ倉庫) に保存
- このフローを GitHub Actions で定期的に全自動で回す(CSS 2025 で発表)

この基盤の利点: 履歴を完全に辿れる



- 手元の検知結果 1 つから、**その時点の上流の姿まで** 戻れる

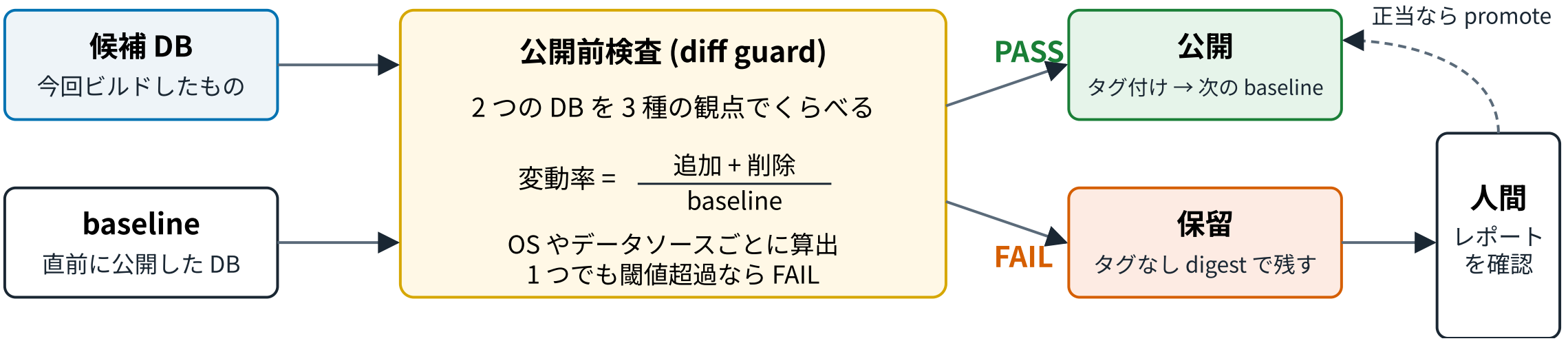
出発点: 2026年3月、壊れた DB を2回公開した

- redhat:9 の検知が **51.9% 消えた** → **見逃し (FN)**
  - Red Hat VEX から壊れたデータが来た日に、こちらが形式を変えた
  - 変換が空振りし、**スカスカのデータ** を生成
- ubuntu 22.04 の検知が **倍になった**(+106.0%) → **過検知 (FP)**
  - Ubuntu CVE Tracker の DB に新しいフラグを入れた
  - **古いスキャナ** がそのフラグを読めない

どちらも上流ではなく **自前のバグ**。それでも **ビルドは成功** し、そのまま公開された。

同じようなバグを、公開前に止める方法が必要。

作ったもの: 公開前検査 (diff guard)



- **検知(新)**: 最新のバイナリで検知実行、CVE の集合をくらべる(検知入力はスキャン結果 75 本)
- **検知(旧)**: 固定した旧バイナリで同じことを。旧版との非互換(3 月の 2 件目)を捕まえる
- **DB 構造**: DB の中の検知条件(約 330 万件)をデータソース別に直接くらべる
- 止めた DB はタグなしで残る。正当なら人が **promote**(公開に昇格)する

実物: 止まったとき、人が見るもの

GitHub Actions のジョブ要約。

3 種の観点のまとめ:

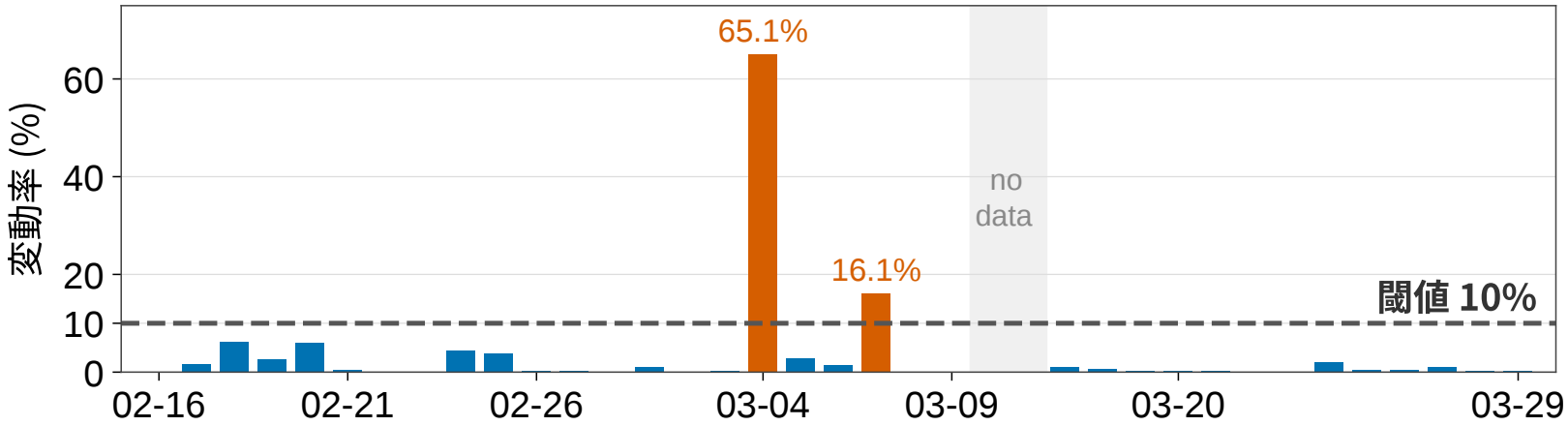
観点	結果
検知(新)	FAIL
検知(旧)	PASS
DB 構造	PASS

検知(新)の内訳(一部):

検知入力	baseline	候補	追加	削除	変動率	結果
windows_11_21h2	879	472	0	407	46.3%	FAIL
windows_11_25h2	1235	722	0	513	41.5%	FAIL
windows_server_2025	1802	1285	0	517	28.7%	FAIL
windows_10_22h2	2697	2198	0	499	18.5%	FAIL

大幅な減少で FAIL。実際には 6 月分の CVE が検知から消えていた。

閾値を決める: 事前に測って、桁の境目でえいや



- 入れる前に、となり合う日の DB 35 組で DB 構造の変動率を測った
- ふだんは中央値 **0.3%**。10% 超は **2 組** だけ(Red Hat の件と、SUSE OVAL の再発行)
- 桁で分かれる境目から閾値を決めた: **DB 構造 10%**、検知は保守的に **5%**

「2 週に 1 回止まる程度かな」と思い、2026-04-23 に本番へ。

現実: 翌日から止まった

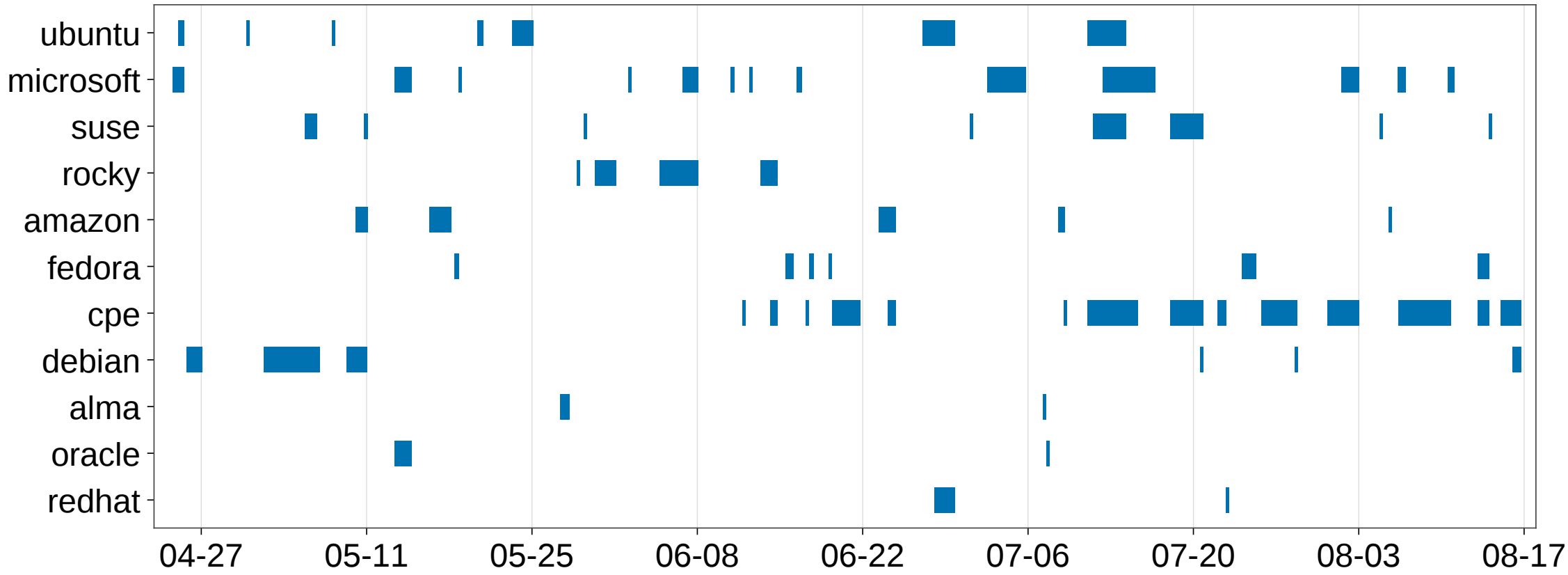
- 翌日(2026-04-24)に止まり、最初の promote(04-27)まで **20 run 連続 FAIL**
- 116 日間で 943 run 中 **420 run(44.5%)が FAIL**。ほぼ毎朝、まず結果を読んで promote
- 止めているあいだは前回版が公開されたまま = **新しい CVE が遅れる**

「2 週に 1 回」どころか、ほぼ毎日。なぜこんなに止まる?

見方を変える: うるさい検査ではなく、上流の地震計

- 「止まりすぎ」を調べるうちに、FAIL の記録は **上流で何が起きたかの記録** だと気づいた
- 2026-04-23 〜 08-16 の **116 日・943 run を全部集めた**
- 同じデータソースの続きの FAIL をひとまとめにすると **69 件**
- 69 件すべての **止まった原因** を特定して、分類する

116日間のタイムライン



69 件をデータソースごとに並べたもの。横棒は最初の FAIL から解消まで

原因の絞り込み: 履歴で工程を消していく

1. **DB 構築**: DB を作ったコードの版をくらべる
2. **収集・変換**: そのデータソースのコードの履歴を見る
3. **収集の設定**: 取る範囲(OS や版)を変えていないか見る
4. **上流**: raw の Git 履歴から、件数と ID が合う変更を探す

全工程の履歴が残っているからできた。去年の基盤がここで効いた。

69 件すべてに原因が付いた。「原因不明」はゼロ。

結果: 69件のうち55件は、上流の正当な変更

原因	件数	例
上流: 正当な変更	55	月例パッチ、新しい OS の立ち上がり
上流: 一時的な障害	2	ひと月分の月例データが消えた(Microsoft)
上流: ずっと続く品質の変化	2	errata の作り直し
自分: 意図した変更	9	CPE まわりの変換、取る対象の追加
自分: 変換コードのバグ	1	出力順が不定

止めるべきもの(上流の障害 2 件)は止められている。
しかし、止めるヒューリスティクスはまだ甘い。

絞り込み実例 (1/2): promote するパターン

```
判定: upstream-driven(上流の正当な変更。promote 可)

FAIL: 検知(新)(旧)で amazon_2023 2080 → 2233(Added 153、7.4%、閾値 5%)

1. builder   : created_by 同一(vuls v0.0.1-alpha.0.20260622074355-fab747f7e5ef)→ 除外
2. extractor : 窓内に pkg/{extract,fetch}/amazon のコミットなし → 除外
3. settings  : vuls-data-db の窓内コミットは他ソースの追加のみ、amazon の設定に変更なし → 除外
4. raw       : vuls-data-raw-amazon cd2dc392(2026-06-23 02:04、185 files / +25,037 行)
   smoking gun: ALAS2023-2026-1882(kernel、Important、Issued 2026-06-22)が新規
                その CVE(CVE-2026-23255, CVE-2026-23272, CVE-2026-23399 …)が Added の ID と一致

推奨: 約 10 日ぶりの advisory 公開でバッチが大きくなっただけ。候補 sha256:d2155fc0… は promote 可(判断・実行は人間)
```

- Claude の SKILL にしてあり、調べる作業の大半は AI エージェントに任せる
- **promote の実行判断は人間**。エージェントに workflow を動かす権限は渡さない

絞り込み実例 (2/2): promote しないパターン

```
判定: upstream-driven(上流の一時的なデータ障害。promote しない)

FAIL: 検知(新)で windows 13 ターゲットが削除のみの大変動
      windows_11_21h2 879 → 472(Removed 407、46.3%)、windows_11_25h2 1235 → 722 ほか

1. builder   : created_by 同一 → 除外
2. fetcher   : pkg/fetch/microsoft/cvrf は期間中コミットなし。fail-hard 設計でスキップ機構も無い → 除外
3. raw       : vuls-data-raw-microsoft-cvrf 4b5baa9 → 44181ca で 2026-Jun/ が丸ごと消失(723 → 0 ファイル)
   smoking gun: 2026-Jun/CVE-2026-48574.json(Windows Media RCE、published 06-09)がファイルごと消失
                extracted 5e47f25 → b21d21d で CVE ファイル 777 本削除(うち 754 が CVE-2026)
                MSRC API は HTTP 200 のまま 6 月の文書を返さなくなったと判断

推奨: 候補 ff6889e1 / b8c1fee3 は promote しない。6 月の Windows 脆弱性を欠いた欠陥品
      上流の回復を待って再実行する
```

- 取得も変換も正常。上流の返答が変わっただけ。候補は捨て、**13 時間後の fetch で戻った**

上流の壊れ方: 消える — Microsoft CVRF

- 6 月分のデータが raw から **丸ごと消え、戻り、また消えた**
- API は HTTP 200。取得も変換も **成功**
- できた DB から 6 月分の Windows CVE が消えた
- 3 回とも、検知(新)が Windows の検知入力 13 本すべてで **減るだけ** の動きを捕まえた

raw の `git log`

日時 (UTC)	変更行数	2026-06 のファイル数
06-15 16:57		723
06-16 03:02	**−251,429**	**0**
06-16 16:33	+248,779	811
…	…	…
07-01 14:17		1273
07-02 02:01	**−327,295**	**0**
07-02 13:26	+335,383	1273
07-03 01:42	**−327,838**	**0**
07-03 13:33	+338,693	1278

上流は骨抜きになる — AlmaLinux errata

- 2026-05-27、AlmaLinux 10 の errata が一斉に減った
  - advisory 自体が **279 → 一時 60 件**(残ったのは 136 件)
  - 残った advisory も骨抜き。例: ALSA-2026:2721(kernel)の対象パッケージが **75 種 → 2 種**
  - **CVE 376 件** が errata 全体から消えた

```
$ git show --stat e64c5ca88 -- 10/ALSA/2026/ALSA-2026:2721.json
 10/ALSA/2026/ALSA-2026:2721.json | 2412 +----------------------------------------
 1 file changed, 6 insertions(+), 2406 deletions(-)
```

- 上流に聞くと「**仕様どおり**」。errata に載るのは **最新リリースのパッケージだけ**。
  10.2 が出て(05-26)、10.1 向けの分が消えていた
- 対応: 落とされた advisory を **raw 履歴から復元** して取り込む

AlmaLinux その2: x86_64_v2 が消えた(2026-10-12) 論文後

- advisory **366 件** から `x86_64_v2` が消えた。`x86_64` は無傷
- rpm 本体は **いまもある**。消えたのは errata の記述だけ
- **DB 構造の観点が捕まえた**
- `x86_64_v2` を見逃す害と、新しい CVE が届かない害を天秤にかけ、**promote した**
- 翌日(10-13)の errata で **元に戻った**。その 1 日、`x86_64_v2` では **検知が消えていた**

例: ALSA-2026:19009 (postgresql18) の packages 数

	x86_64	x86_64_v2
10-12 以前	13	13
10-12 20:40Z	13	**0**
10-13 18:35Z	13	13

上流は改善をやめない — VulnCheck

- **1 コミットで 145,672 ファイル**。全年代の CVE に CPE 条件を一斉付与
- **DB 構造の観点** が FAIL(cpe 全体で 28.9%)。「増やしただけだろう」と見て **promote した**
- 実際は CPE の書き方も変わっていて、CPE 経由の検知が **最大 75% 消えていた**

```
例: CVE-2012-5010(Cisco ASA)の CPE 条件
  07-08 より前   cpe:2.3:a:cisco:adaptive_security_appliance_software:-:*:*:*:*:*:*:*
  07-08 以降     cpe:2.3:o:cisco:adaptive_security_appliance_software:*:*:*:*:*:*:*:*
```

- 「増えた」の陰に「検知が減った」が隠れていた。**止めたが、人が通した**

上流は改善をやめない(2): VulnCheck の10万ファイル往復

- 前の枚の一斉付与(07-08)のあと、**約 10 万ファイルが 12 時間ごとに 2 つの版を行き来** した
- 07-21 に `a:cisco:…` の CPE が 82 ファイルに現れ、翌日消えた。
  検知が **22 → 67 → 22** と 7 時間で往復
- cpe_cisco の VulnCheck 由来の検知は **67 → 10 件**(2026-10-14)。戻らないまま

日時 (UTC)	変更ファイル数	補足
07-08 13:24	145,672	一斉付与(前の枚)
07-10 13:53	101,264	別の版に書き換え
07-11 01:21	99,986	**07-08 の版に戻る**(抽出 3,000 ファイルで 100% 一致)
07-11 12:48	99,196	**07-10 の版に戻る**(同上)
07-21 13:12	5,925	`a:cisco:…` が 82 ファイルに出現
07-22 01:19	3,579	その `a:cisco:…` が消える

現在地(1): 公開前検査が止めた3件

- **Microsoft CVRF の 6 月分の消失 × 2**:
  Windows の 6 月分 CVE(最大約 1,270 件)の検知が消える DB を **公開せずに捨てた**
- **自分の変換コードのバグ × 1**(fedora):
  検知には効かないが、**出力が揺れる** バグ。
  テストでは見つからず、揺れは履歴の調査も邪魔する。出力の差分で初めて見えた

現在地(2): 公開前検査が止めきれなかった2件

- **AlmaLinux の骨抜き**:
  連続 FAIL を片づける流れで promote → **数時間公開してしまった**
- **VulnCheck**:
  DB 構造の観点は cpe 全体(7 データソース合算)で **28.9%**。単体なら **189.6%**
  薄まった数字を見て promote → 検知が減った DB を公開した
- 対策(2026-07-14): **データソースごと** に測る

2 件とも、見つけて止めるまではできた。
見せ方と人の判断は、まだ甘い。

まとめ

1. 脆弱性情報の上流は 2 日に 1 件のペースで **大きく動く**。
   消え、戻り、書き換わる。**大半は正当な変更**。
2. でも、**本当にまずい DB** がたまにできる。
3. 公開前に **前回の DB** との差を見て、**大きな差分** はひとまず止めるようにした。
   本当にまずい DB を 2 回捨てた、少しまずい DB も 1 回捨てた。

今後の針路

- **止まりすぎ** → 増える方向と減る方向で閾値を分け、**減る方向を厳しく** (論文後に実装済み)
- **止めると遅れる** → **データソース単位の据え置き**: だめな分だけ前回版で公開(実装中)
- **人の判断** → 検査ができるのは自動で止めるまで、**人間系の設計** は長期的課題

公開しています

論文・データ(run 表、69 件の判定表、集計スクリプト)

github.com/vulsio/css2026-diff-guard

検査の実装: github.com/MaineK00n/vuls2

閾値の設定: github.com/vulsio/vuls-data-db

以降は予備スライド

予備: 上流の一時的な欠落 → promote しない(2026-09-26)

```
判定: upstream-driven(JVN フィードの一時的な欠落。据え置き推奨)

FAIL: cpe_nvd / jvn-feed-rss  71 → 56(Removed 15、21.1%、閾値 5%)

1. builder   : created_by 同一 → 除外
2. extractor : extractor・fetcher は 09-20 以降コミットなし → 除外
3. settings  : vuls-data-db は 09-24〜27 コミットなし → 除外
4. raw       : vuls-data-raw-jvn-feed-rss d37d1eda43 → ac66be6273 は A:490 / M:1 / D:426(前日は D:0)
   smoking gun: 削除 426 件は全件 modified が 2026-09-25。JVNDB-2026-034354〜034784 の連番ブロック
                代表 2026/JVNDB-2026-034784.json(Apache Tomcat)が削除 → 参照する 15 CVE が検知から消えた
   原因: jvndb_new.rdf は直近 500 件の窓。9/25 は 1224 件公開され前半 426 件が押し出され、
         年次 jvndb_2026.rdf にまだ載っていない隙間に fetch が当たった

推奨: 閾値変更は不要。promote もしない。候補 DB は 426 件を欠いた状態なので据え置きが正解
      次回 fetch で raw が復元され、次の DB run は Added 側に振れて PASS するはず
```

- 取得側は正常。「窓 500 件あれば年次の遅れを埋められる」という **決めつけが破れた**

予備: 後日、データソース単位で測り直すと(VulnCheck)

同じ 2 つの DB を **DB 構造の観点** でデータソースごとに測り直す

ecosystem	source	change rate	result
cpe	nvd-api-cve	≤ 5.4%	PASS
cpe	jvn-feed-detail	≤ 5.4%	PASS
cpe	cisco-json	≤ 5.4%	PASS
cpe	vulncheck-nist-nvd2	**189.6%**	**FAIL**

検知の観点 に CPE の検知入力を足して再実行 (検知した CVE 数)

scan result	baseline	target	変化
cpe_jvn	69	17	**−75.4%**
cpe_cisco	67	24	**−64.2%**
cpe_nvd	24	19	−20.8%

- 28.9% の正体は **VulnCheck ひとつで 189.6%**。NVD の大きさに埋もれて薄まっていた
- CPE 経由の検知が **60〜75% 消えていた**。書き方が変わり、索引から外れた
- 「増えた」と「検知が減った」が同時に来る。公開した DB には検知の後退が入っていた
- → 閾値を **データソースごと** に。CPE の検知入力も検知の観点に足した(2026-07-14)

予備: 観測が終わった4日後、Fortinet

- advisory のページが、誰にでも Cloudflare の確認画面を **HTTP 200** で返すようになった
- 取得側は「CSAF なし」と判断し、何も書かず **exit 0** → 「上流が取り下げた」として **削除を記録**
- 2026-01 から 1 件ずつ消え、**08-20 に残り 181 件を全部削除**。run は緑。次の run が空で落ちて気づいた
- 対策: 23 本の取得 workflow に「**空になった repo は記録しない**」

```
$ git log --diff-filter=D   (vuls-data-raw-fortinet-csaf)
f4f1d3d 2026-01-22  deleted   2   left  88
2dd3ea9 2026-02-06  deleted   1   left  89
d1ed230 2026-02-21  deleted   1   left  98
c79f291 2026-02-28  deleted   1   left  97
154c743 2026-03-08  deleted   1   left  96
ac1fda0 2026-04-07  deleted   1   left 119
25f9f6c 2026-04-29  deleted   1   left 146
5a8924a 2026-06-28  deleted   1   left 162
56fc622 2026-08-20  deleted 181   left   0
        181 files changed, 59632 deletions(-)
```

```
$ curl -i https://fortiguard.fortinet.com/psirt/FG-IR-25-250   (2026-10-09)
HTTP/2 500
<title> The URL you requested has been blocked </title>
```

予備: 取得の段階で見える壊れ方(2026-09〜10)

DB に入っているデータソース

- **Fedora**: Bugzilla が security bug を `NotPermitted` で返す(**5.4 万 → 5.9 万件**)。CVE ID が advisory から消え、fedora_42 が 13.5% で FAIL
- **ENISA EUVD**: 201 ページ目で `403 Forbidden`。取得が止まり、更新も止まる

取っているだけのデータソース(DB には未収載)

- **Wind River**: **穴埋め前のひな形の行** がそのまま配信 (17 ファイル)
- **openEuler**: CSAF の `product_ids` が配列ではなく object

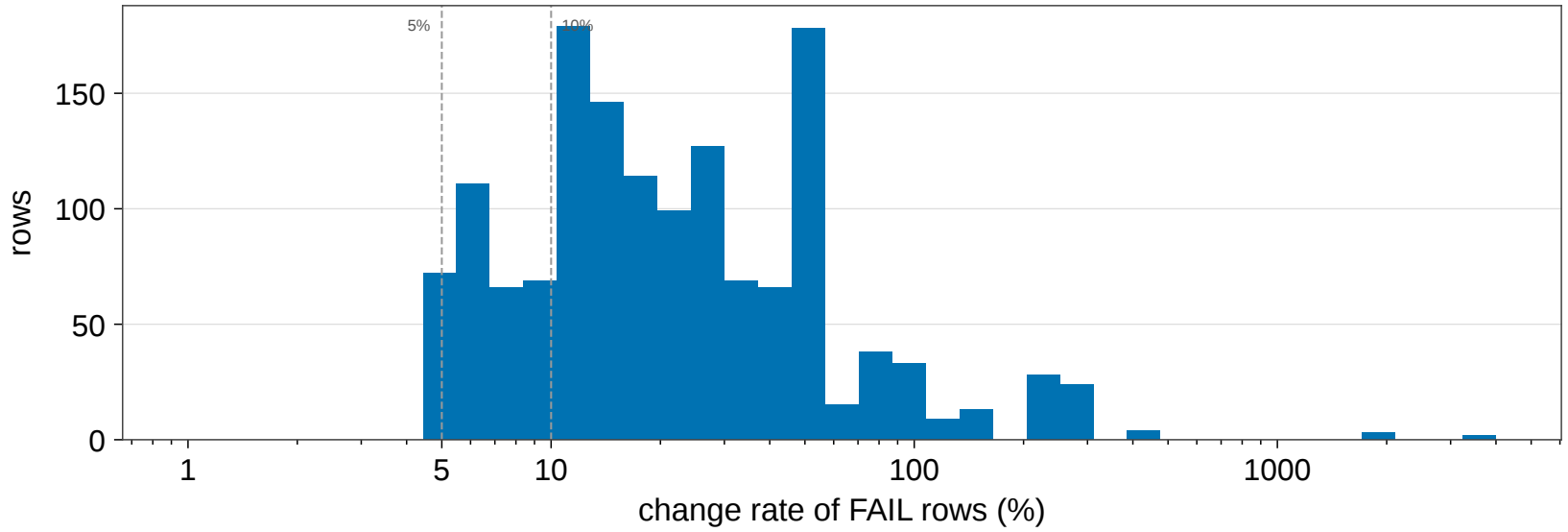
```
# fedora-api raw(2026-09)
{"bug_id": "2398586", "error": "NotPermitted"}
# REST: code 102 "You are not authorized to access bug #2398586"
```

```
# enisa-euvd fetch(2026-10-07)
GET .../api/search?page=201&size=100
  giving up after 6 attempt(s): unexpected HTTP status 403 Forbidden
```

```
# windriver-cve-tracker  active/CVE-2010-5312
Patches_chromium-x11:
{recipe_rcpl}_{package}: {status}
```

```
# openeuler-csaf fetch(2026-10-07)
json: cannot unmarshal JSON object into Go []csaf.ProductID
```

予備: 正当でも桁違いに動く



- 最大は fedora:46 の **3433.3%**: 末端条件が **6 → 212 件**。異常の大きさではなく **分母の小ささ**
- Oracle UEK は **1 つの advisory に CVE 822 件**。kernel CNA の一括発番(2026-07-24〜25)は Debian と VulnCheck に同じ日に流れ込んだ
- ふだんの 5〜20% と、100% 超の大きな出来事が裾を作る

予備: 上流は黙って書き換わる — Cisco openVuln API

- Cisco の advisory には **影響する製品・版の一覧**(productNames)がある。2026-07 なかばから、これが **毎日 12:45Z ごろ黙って書き換わる**
- 例: 2014 年の IOS SIP の advisory。影響する版が **149 → 36**
  - 消えたのは `IOS 15.3(3)JN14` など、**2020 年代の AP 用 IOS の版** 113 件
  - advisory の **lastUpdated(2014-03-31)も version も status も変わらない**
- 整理には見える。でも AP や WLC を CPE で見ている人には **検知が消える**。お知らせはない

```
$ git show f5b5023d -- 2014/cisco-sa-20140326-sip.json          # 2026-07-24
   "productNames": [
-    "Cisco IOS 15.3(3)JN14",
-    "Cisco IOS 15.3(3)JN15",
     ... 113 の版を削除 ...
   "lastUpdated": "2014-03-31T13:46:00",  "version": "1.1",  "status": "Final"   ← 変わらない
```