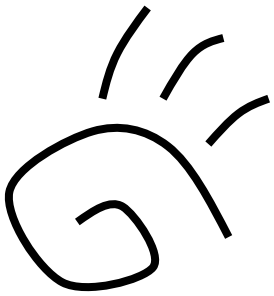


ぼくの考えたさいきょうのDevSecOps

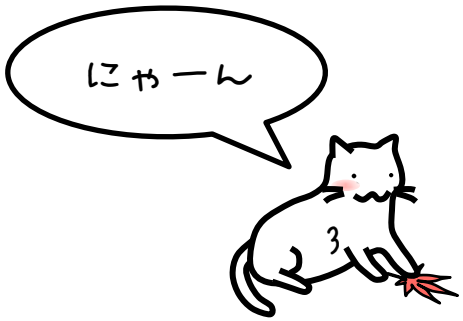
2017/1/19

Vulsといかれたメンバー

自己紹介するよ！

- kotakanbe
- yujiro
  - フューチャーアーキテクトっていう会社で未来を作っているよ
- momo
  - アイアイジェイっていう会社でアイを唱えているよ
- Thetakei
  - エヌ・ティ・ティソフトウェアというやわらかな会社だよ

はて



「セキュリティは経営課題！」

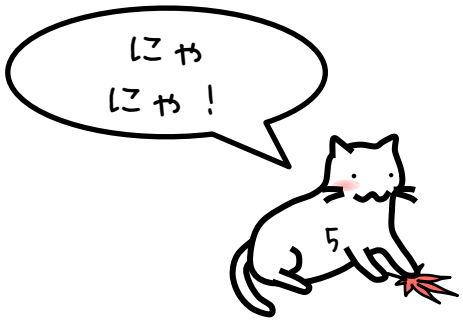
» 脆弱性が出ると「すぐにパッチを当てろ」とのお達しが

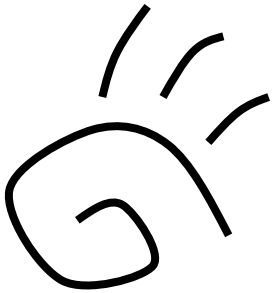
× 去年はBINDが当たり年！（いや、困るよ……）

» どこまで影響出るんだよ……時間かかるよ……　ざわ……ざわ……

» 「その間どうやって守るんだよ！？」とか言われても……

さらに





NW機器も脆弱性が！

» 脆弱性はサーバだけじゃない！

» ルータやFW, UTMなど外側のNW機器にも！

[redacted]に情報漏洩の脆弱性 - 「Shadow Brokers」のエクスプロイト調査中に判明

[redacted]のネットワーク機器向けソフトウェア[redacted]「同XE」「同XR」に、情報漏洩の脆弱性が存在することが明らかとなった。同社はアップデートの準備を進めている。

これらソフトウェアの一部バージョンに、「IKEv1」のパケット処理においてメモリの内容が取得される脆弱性「CVE-2016-6415」が存在することが判明したもの。リモートより認証なく攻撃が可能で、重要度は、4段階中上から2番目にあたる「高（high）」にレーティングされている。

影響を受けるソフトウェアを搭載した機器において、「IKEv1」のネゴシエーションを受け入れる設定の場合、細工したパケットを利用して、機密情報を取得されるおそれがある。

ZDNet Japan

編集部からのお知らせ
• 海外ニュース記事ライター（フリーランス）募集

CIO | クラウド | モバイル PC | セキュリティ | ソフトウェア | サーバ | ストレージ | ネットワーク | ミドルウェア | 運用 | マーケティング

ZDNet Japan > セキュリティ

脆弱性が発見された[redacted]の「[redacted]」にデフォルトのバックドアパスワード

Chris Duckett （ZDNet.com） 翻訳校正： 編集部 2015年12月22日 11時02分

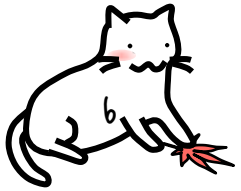
Tweet B!

印刷 | メール | ダウンロード | クリップ

先週脆弱性が判明した[redacted]nOS」にデフォルトのパスワードが存在することを、セキュリティ企業Rapid7が明らかにした。

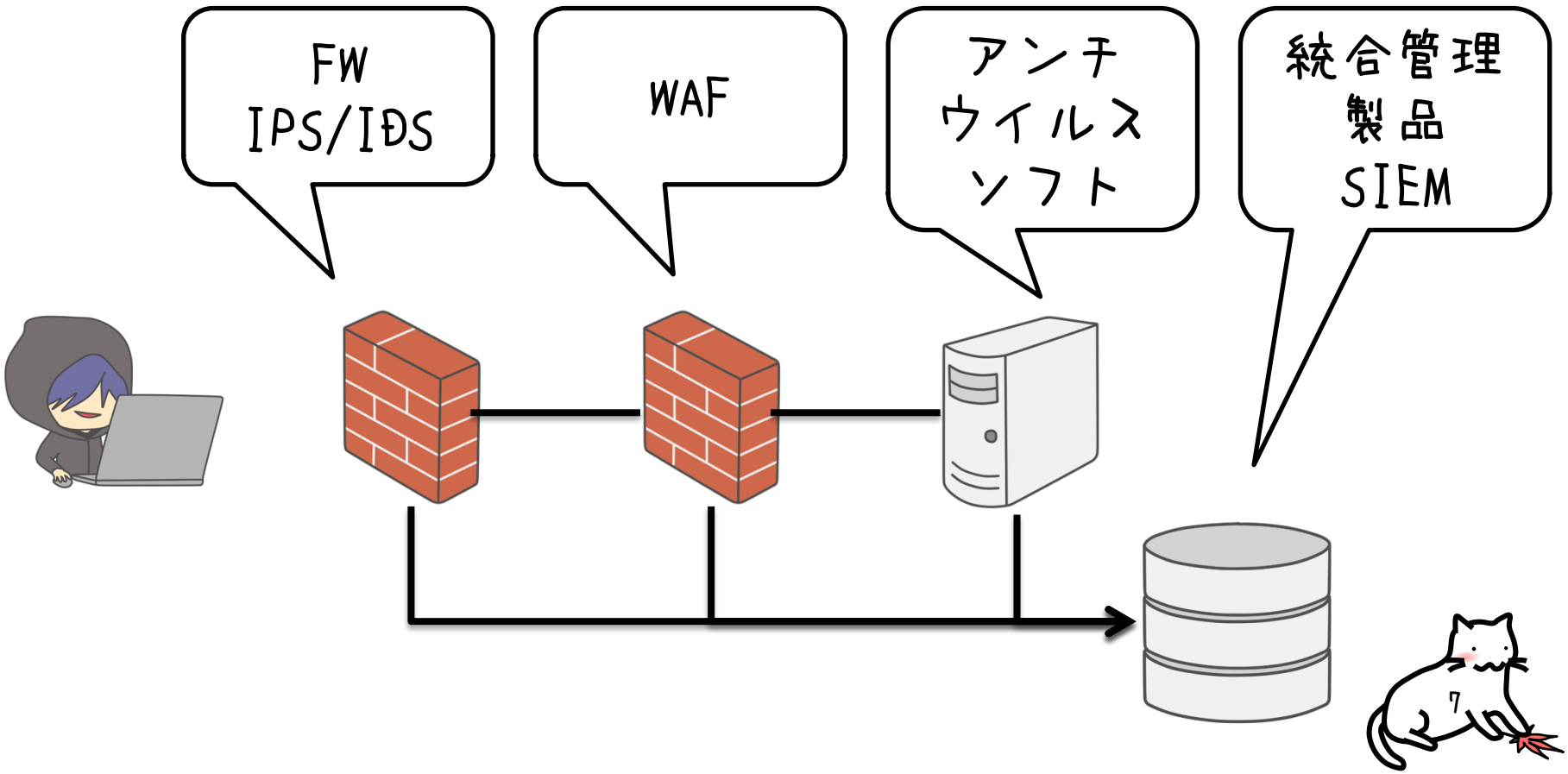
Rapid7の最高セキュリティ責任者、HD Moore氏はブログで[redacted]のパッチが当てられたバージョンと脆弱なバージョンとの違いをstrcmp呼び出しにフォーカスして調べていて、パスワードの存在に気付いたという。パスワードは<<< %s(un='%s') = %uだ。これを知っていれば、有効なユーザー名があればログインが可能になるという。

[redacted]は先週、内部のコードレビューにより[redacted]に2件の脆弱性を発見したことを報告した。1つ目は攻撃者が足跡を残さずにVPNトラフィックを復号するこ



お金の力で！

» お金で緩和する方法もとれるかも？



本当に！？

» そんなの買う予算とってないよ！

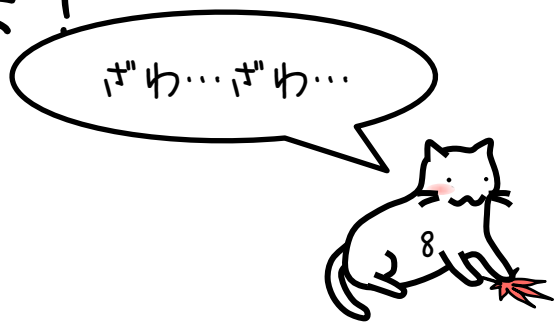
× 今年度の予算はもうない！

» それじゃ止まらないよ！

× だよねー

» 入れても運用できる人がいないよ！

× だよねー



じゃあどうしよう？

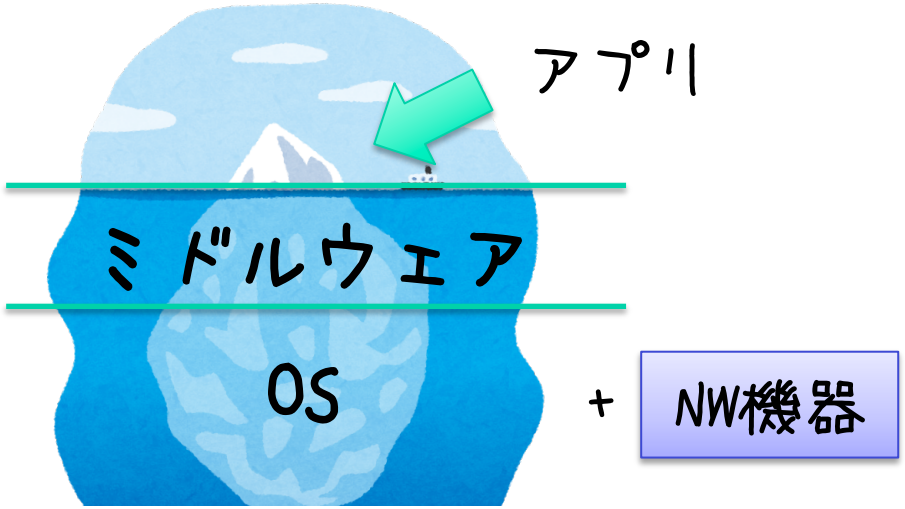
» 放置はできない…

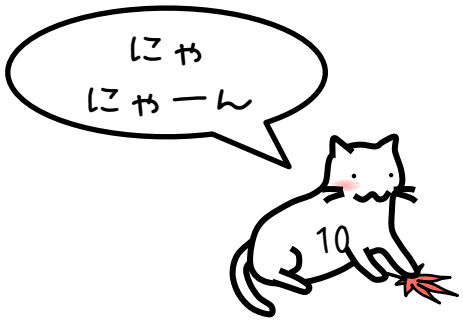
» 本来はちゃんとパッチを当てて適切に設定できているかどうか、のはず

» 0円でOSSで自動でリリース判定までできて可視化できたらいいなあ

そこで奥さん、DevOpsですよ

- DevOpsとは（略
- いやいや、セキュリティ要素ゼロですよ！
- NW機器まで見てますか！？
- 自分のアプリだけじゃないですよ！

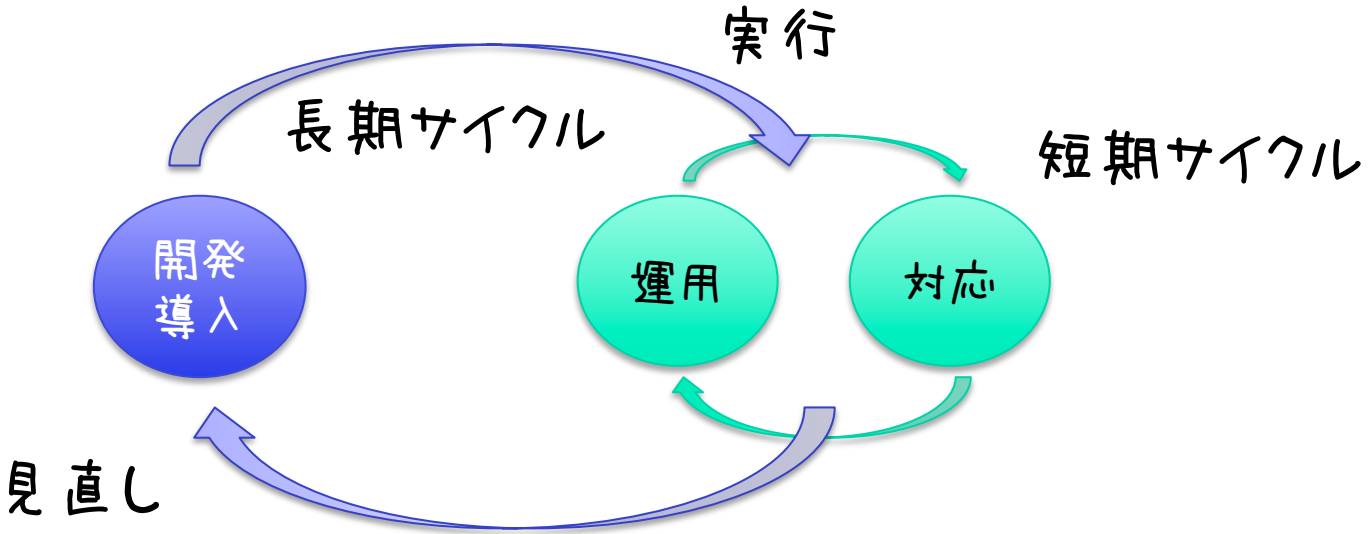




じゃあ何するのさ！？

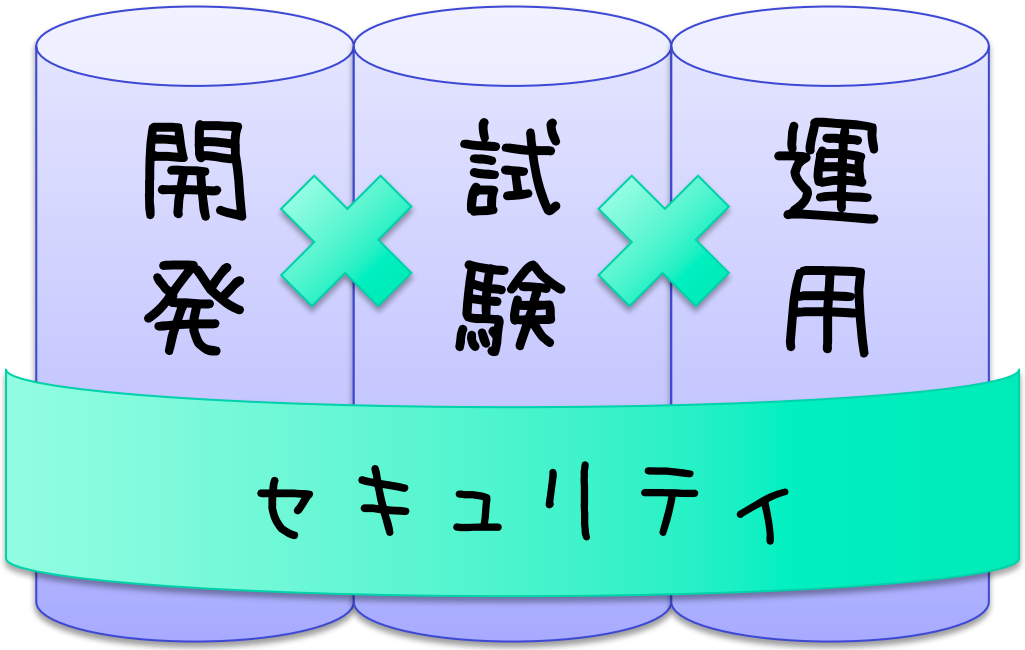
» ここでは「予防」としてのパッチの適用を対象とします。

» 短期サイクルの「運用」と「対応」ができるよう、迅速なリリースを目指します



ありたい姿

» 「開発」、「試験」、「運用」が一体でセキュリティも含めて取り組む



今日議論したいDevSecOpsとは

» 脆弱性が出たらパッチを適用

» 全レイヤーでの試験を実施し、リリース判断

» 関係者全員が一目瞭然で状況が見える

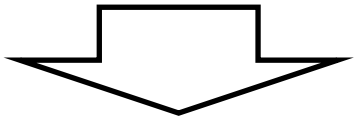
» そんなこといいな、できたらいいな

「試験」とは

- 試験の観点は2種類。
- 内側から：最新かどうか。設定に抜け漏れはないか。パッチの影響はないか。
  - × ホワイトボックス的に。
- 外側から：期待通りの設定や動作になっているか。スキャンしてチェック。
  - × ブラックボックス的に。
  - × これまでの脆弱性診断もこちらに近い（かも）

どこまでが範囲？

- 開発が中心だとサーバに閉じる
- NW機器の外側からスキャンすれば、NW機器の設定も確認できる



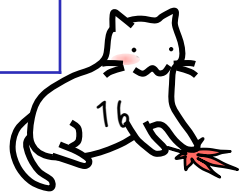
- NW機器のバージョンや設定も確認できれば、試験範囲に入れられる！

組み合わせてみよう！OSSで！

» Vulsを中心に既存のツールと連動

» 可視化にはfaradayも活用する

対象	内側から試験	外側から試験
アプリケーション	(UnitTest)	OWASP ZAP
アプリケーションライブラリ	Vuls + OWASP Dependency Check	
ミドルウェア	Vuls	Nmap, Nessus, Metasploit, OpenVAS
OS		
NW機器	Vuls + OVAL	
可視化	Vuls repo or faraday	

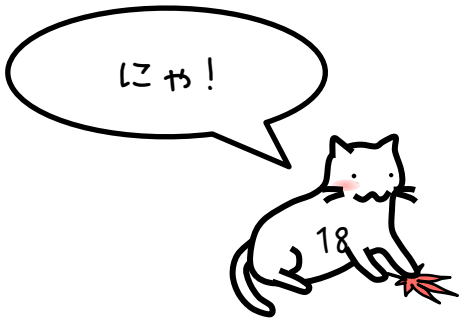


（参考）ツール紹介

- Faraday
  - https://www.faradaysec.com/
  - 各種スキャン結果などを取り込んで統合的に可視化するツール。コミュニティ版は無償。
- OWASP Dependency Check
  - https://www.owasp.org/index.php/OWASP_Dependency_Check
  - 利用しているコンポーネントの依存関係から脆弱なものを使っていないかチェックする
- OWASP ZAP (Zed Attack Proxy)
  - https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
  - Webページの脆弱性診断で利用される。ローカルプロキシとして動作し自動でチェックしたりパラメータを変化させてチェックなどできる
- Nmap
  - https://nmap.org/
  - ネットワークスキャナ
- Nessus
  - https://www.tenable.com/products/nessus-vulnerability-scanner
  - 脆弱性スキャナ。商用では有償。個人向けfor Homeは無償。
- OpenVAS
  - http://www.openvas.org/
  - 脆弱性スキャナ。
- Metasploit
  - https://www.metasploit.com/
  - 脆弱性診断ツール。脆弱性のスキャンに加えて、攻撃コードを利用した診断もできる。

まてまてーい！

vulsだよ！

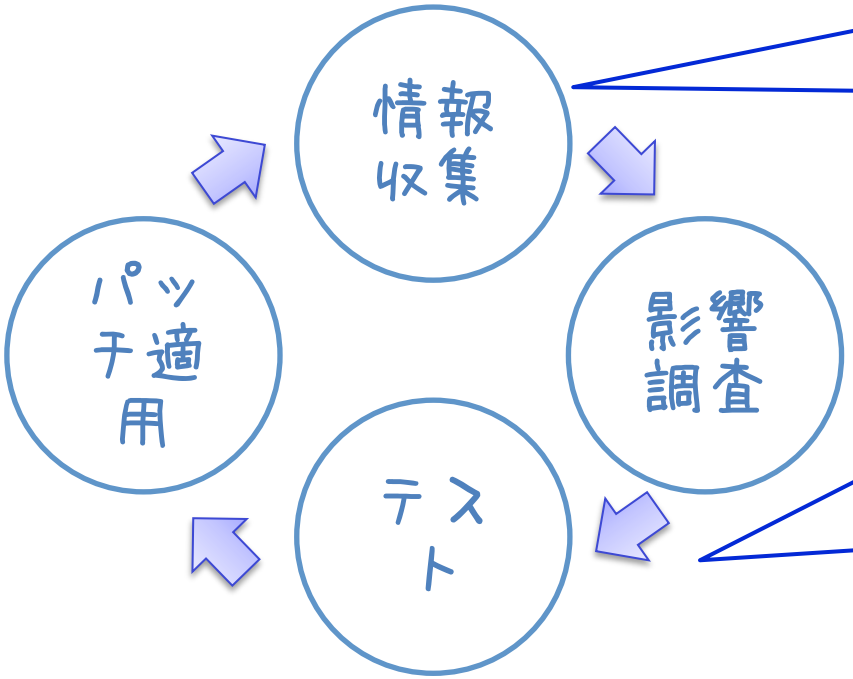


IDS/Antivirusでは防げない脆弱性のデモ

- Dirty Cow (CVE-2016-5195)
- SSH接続可能な一般ユーザがRootへ権限昇格可能な脆弱性
- 内部犯行可能
  - SSHで接続可能なユーザであれば誰でも簡単に実行可能
  - 運用ルールで一般、rootユーザを分けていても意味なし
  - PoCが沢山公開されているので誰でも簡単に実行可能
  - https://dirtycow.ninja/
- 根本的な解決はソフトウェア・アップデート

手動アップデート運用での脆弱性対策は大変

» 日々公開される脆弱性情報を追い続ける必要がある
» システム運用者は以下のサイクルを日々回している



- 約7千件/年の脆弱性
- 全部チェックするのは大変
- 漏れたら放置されてしまう

- 大規模、カオスな環境は大変
- パッケージ、ライブラリ、N/W機器
- 漏れはないか？

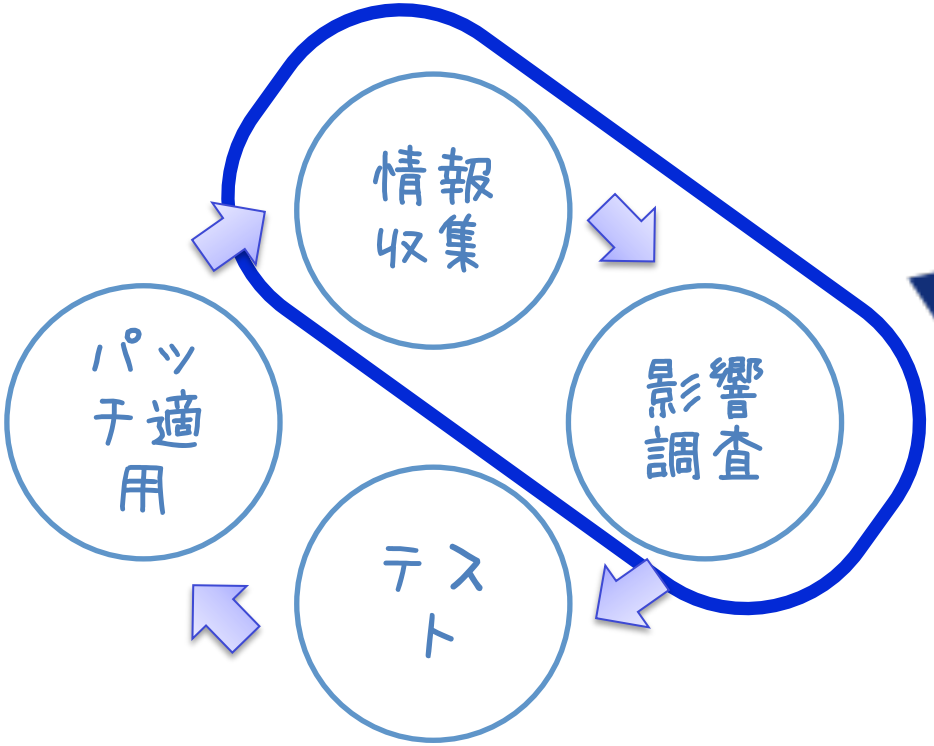
毎日の大変な作業は自動化すべき

脆弱性の分類

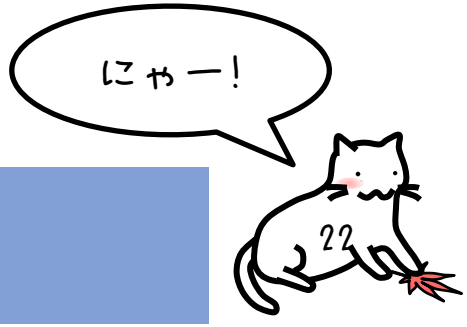
- 脆弱性対策にまず必要なのは、
  - × 「Visibility（可視化）」と、可視化したセキュリティ状況の
  - × 「モニタリング」
  - × 「見えないことは、コントロール出来ない」
- 近年、脆弱性の種類も多様化し、一つの対策方法だけでは防げない。

Vulsは脆弱性情報の収集、影響度調査を自動化する

» 日々の情報収集、影響度調査を自動化
» パッチ適用漏れによる脆弱性の放置を防ぐ



Vuls

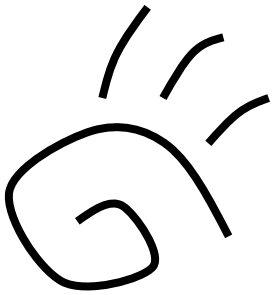


私が作りました

Vuls 概要

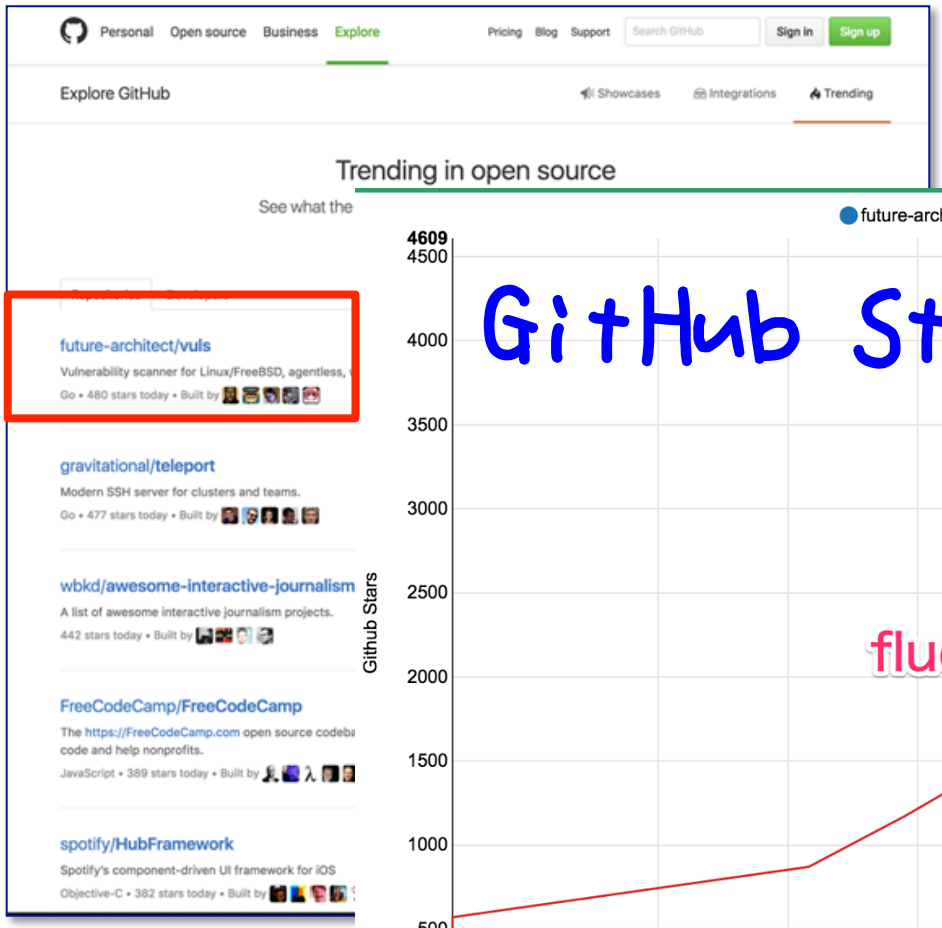
» 日々の脆弱性対策作業を自動化

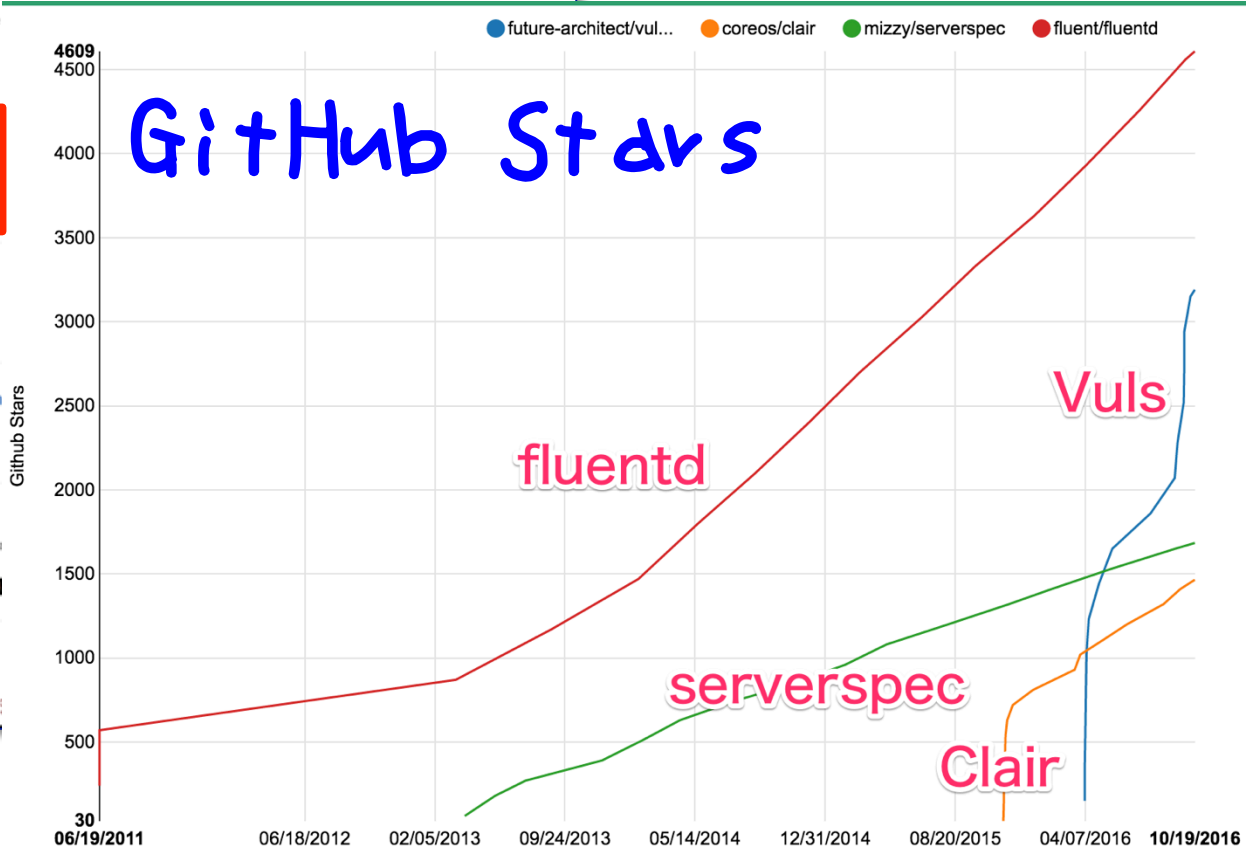
» 潜在する脆弱性と該当サーバを可視化



Vulsの盛り上がり

一時 1 位 (/約3,000万) に！





Vulsが検知可能な脆弱性

» CVE-IDが割り当てられているもの

» OSパッケージマネージャ管理下のソフトウェア

× アップデートしたら解決するCVE-IDが割り当てられた脆弱性

» コンパイルして入れたソフトウェア

× CPEに登録されているもの、かつ、CVE-IDが割り当てられた脆弱性

» プログラミング言語ライブラリの脆弱性

× CPEに登録されているもの、かつ、CVE-IDが割り当てられた脆弱性

できないこともある！

» ゼロデイ脆弱性

» CVE-IDは割り当てられているが、パッチがまだ公開されていない脆弱性

» 自分でプログラミングした部分に含まれる脆弱性

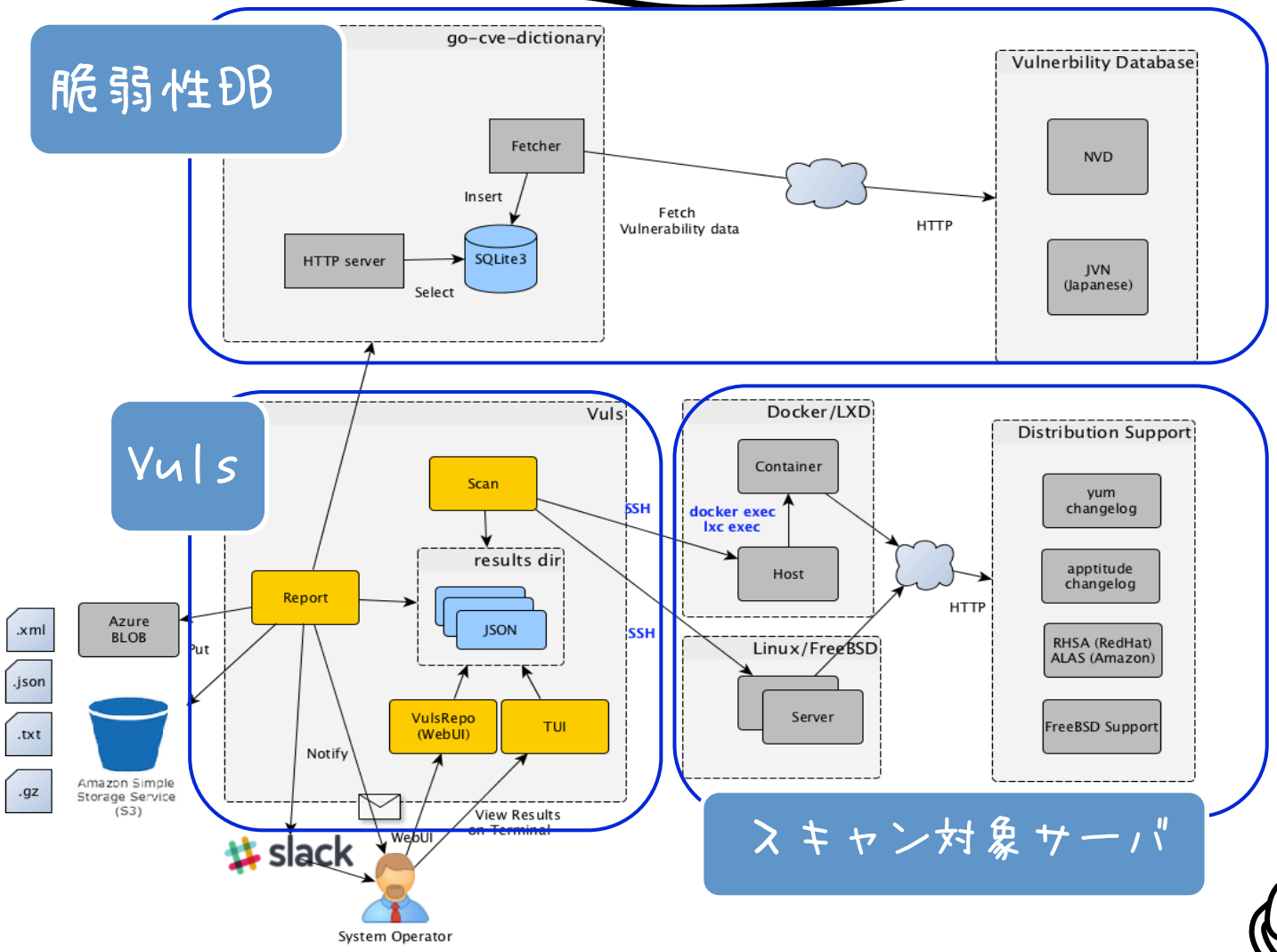
× Web脆弱性など

2つのモードで動作可能

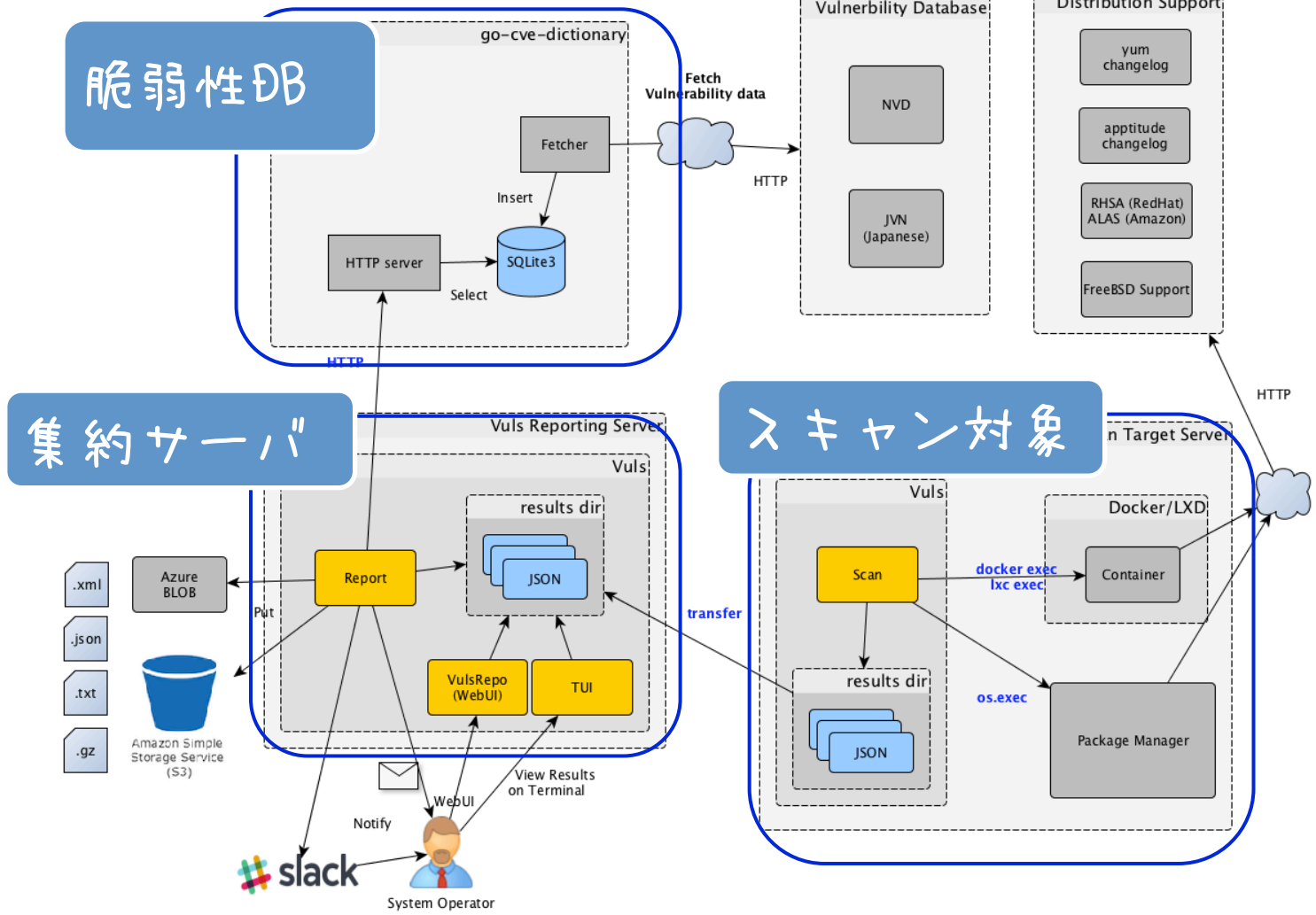
» 中央からSSHで接続してスキャン

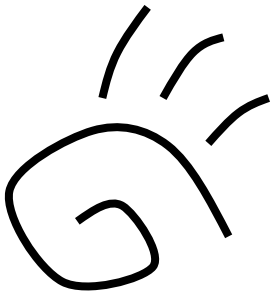
» 各サーバにVuls配置（ローカルスキャン）

中央からSSHで接続してスキャン

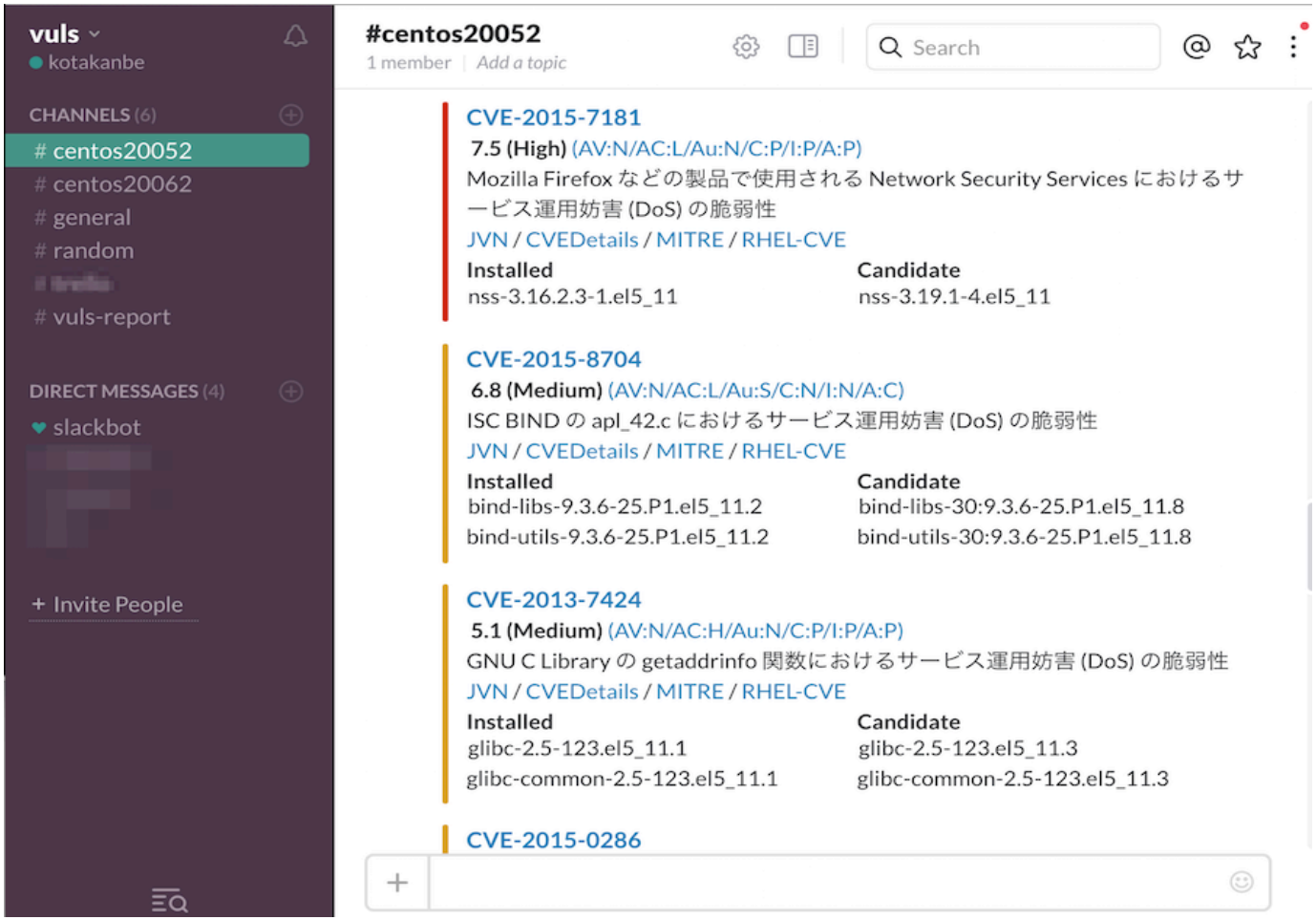


各サーバにVuls配置（ローカルスキャン）



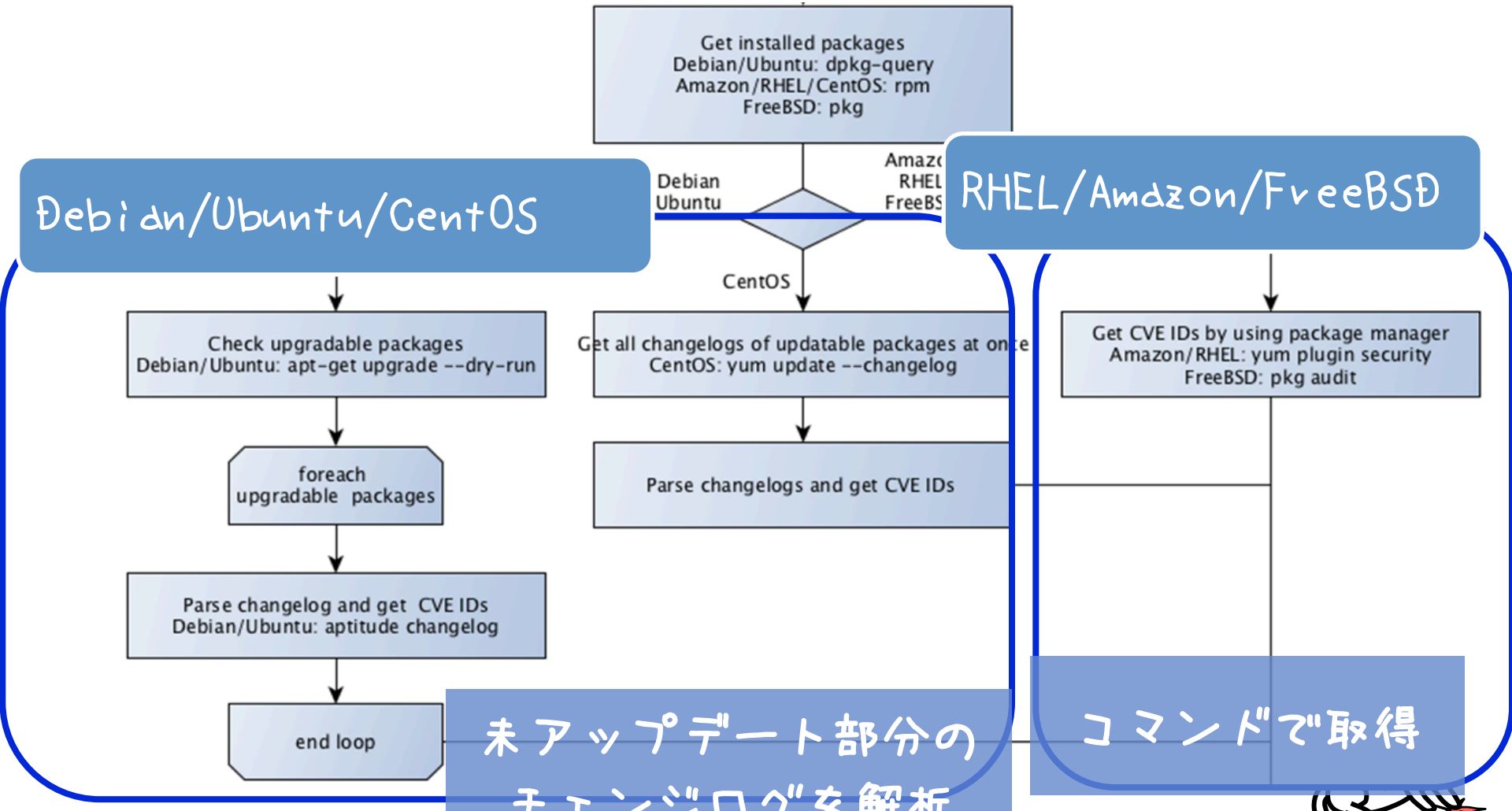


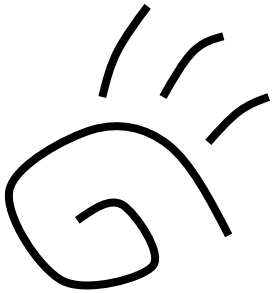
デモ


vuls
kotakanbe
CHANNELS (6)
centos20052
centos20062
general
random
vuls-report
DIRECT MESSAGES (4)
slackbot
+ Invite People
#centos20052
1 member | Add a topic
Search
CVE-2015-7181
7.5 (High) (AV:N/AC:L/Au:N/C:P/I:P/A:P)
Mozilla Firefox などの製品で使用される Network Security Services におけるサービス運用妨害 (DoS) の脆弱性
JVN / CVEDetails / MITRE / RHEL-CVE
Installed
nss-3.16.2.3-1.el5_11
Candidate
nss-3.19.1-4.el5_11
CVE-2015-8704
6.8 (Medium) (AV:N/AC:L/Au:S/C:N/I:N/A:C)
ISC BIND の apl_42.c におけるサービス運用妨害 (DoS) の脆弱性
JVN / CVEDetails / MITRE / RHEL-CVE
Installed
bind-libs-9.3.6-25.P1.el5_11.2
bind-utils-9.3.6-25.P1.el5_11.2
Candidate
bind-libs-30:9.3.6-25.P1.el5_11.8
bind-utils-30:9.3.6-25.P1.el5_11.8
CVE-2013-7424
5.1 (Medium) (AV:N/AC:H/Au:N/C:P/I:P/A:P)
GNU C Library の getaddrinfo 関数におけるサービス運用妨害 (DoS) の脆弱性
JVN / CVEDetails / MITRE / RHEL-CVE
Installed
glibc-2.5-123.el5_11.1
glibc-common-2.5-123.el5_11.1
Candidate
glibc-2.5-123.el5_11.3
glibc-common-2.5-123.el5_11.3
CVE-2015-0286

30

OSごとのスキャンフロー





今後はスキャン精度の向上に取り組む

» 脆弱性情報の情報ソースをリッチに

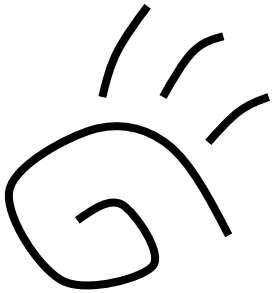
× 現在はNVDとJVNのみだが、ディストリビューション特有の脆弱性情報も取り込む

» 複数のスキャン方法で検知

× OVALを使った検知

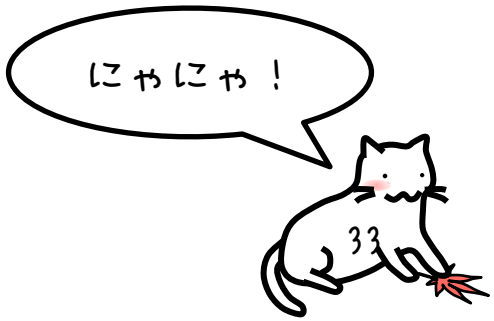
× チェンジログパースでの検知と組み合わせる

» 検知精度でナンバーワンを目指す！



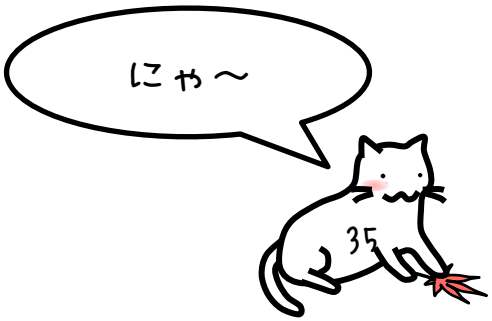
まだまだー！

OVALだよ！



話の流れ

- 発端
- OVALって何？
- やってみた
- これからどうする？



※ねこと内容はまったく関係ありません

発端

» JANOG38沖縄、セキュリティツールBoFにて…

» Vulsと世界平和について大いに語った
(@kotakanbe)

× その後モヒカンSlackも話題になりましたよ

» BoF終了後、Ciscoの方から…

× 「ネットワーク機器も見て欲しい」

× 「OVALってのを出してまして(以下略)」

× なにそれ、使いたい！

OVALって何？

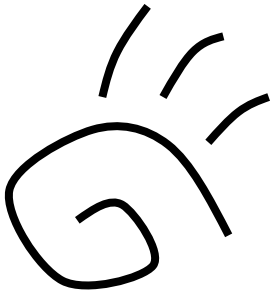
» IPAさんが解説してくれてます

× セキュリティ検査言語OVAL概説
https://www.ipa.go.jp/security/vuln/OVAL.html

× OVAL（本家） MITREから移ってます
https://oval.cisecurity.org/
http://oval.mitre.org/

» 検査手順を記述するための言語（XML）

× つまり…プログラムで自動化しやすい！



やってみよう（下調べ）

» OVAL定義の元ネタ：OVAL Language

https://github.com/OVALProject/Language

README.md

OVAL Language

International in scope and free for public use, OVAL is an information security community effort to standardize how to assess and report upon the machine state of computer systems. OVAL includes a language to encode system details, and an assortment of content repositories held throughout the community.

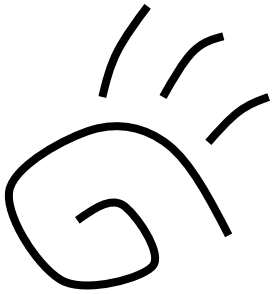
This GitHub project hosts the OVAL Language Schemas and Specifications to improve transparency and the community's ability to track changes before a release. As of the OVAL 5.11.1 release, official releases of the OVAL Language are posted on this GitHub project.

Development: OVAL 5.11.2
Official: OVAL 5.11.2
Previous: OVAL 5.11.1

Experimental capabilities for the OVAL Language will be developed in the OVAL Language Sandbox to allow the community to fully investigate and implement new capabilities before they are included here in an official release ensuring that only mature and implementable constructs are added to the OVAL Language.

Please see the OVAL Web Site for more information about the OVAL Language.

The OVAL Language operates under the OVAL Terms of Use.



やってみよう(下調べ)

» OVAL Interpreterがあるらしい…

(Windows版はインストールできました)
(けど使ってみてません)

Home / Browse / Security & Utilities / Security / OVAL Interpreter / Files

OVAL Interpreter

Brought to you by: bob_roberge, bworrell, chishomi, djhaynes, and 4 others

Summary | Files | Reviews | Support | Wiki | Tickets | News | Discussion | Code

Looking for the latest version? Download ovaldi-5.10.1.7-src.tar.bz2 (20.4 MB)

Home / ovaldi

Name	Modified	Size	Downloads / Week
Parent folder			
5.10.1 Build 7	2014-04-10		45
5.10.1 Build 6	2014-01-04		0
5.10.1 Build 5	2013-04-17		0
5.10.1 Build 4	2012-12-21		0

やってみよう(下調べ)

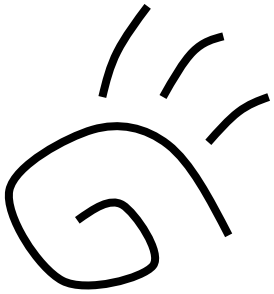
» github を検索した。別実装があるようだ。

(売り物？
スキャンだけしたい…)

	Developer Edition Embeddable SCAP engine for software companies	Enterprise Edition Scalable SCAP scanning for organizations and service providers	Professional Edition Desktop SCAP scanner for content authors and small, ad hoc scans
Basic Features			
Included Components	**Java SDK** 100% pure Java library for scanning **Joval Sensor** Server-based component for high-volume agentless scanning **Joval Utility** Lightweight command-line utility for local scanning		Desktop application

やってみよう(下調べ結果)

- 使いたいのは一部だから、作っちゃおう！
  - 今回やってみたいのはCiscoだけ
  - 作った方が後からVulsに組み込みやすそう？
- XMLを読み込んで、その通りに検査する
  - LLでプロトタイプ実装



やってみよう(Ciscoの配布物)

» Ciscoの提供しているOVALを覗いてみる

https://tools.cisco.com/security/center/ovalListing.x

Worldwide [change] Welcome, | Account | Log Out My Cisco

cisco Products & Services Support How to Buy Training & Events Partners

Home / Cisco Security / Latest Threat Information / Cisco Security Advisories and Responses

Cisco Security

OVAL Repository

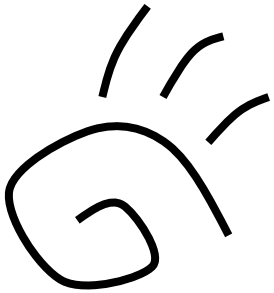
Cisco Security Advisories pertaining to Cisco IOS Software are available below in OVAL Definition schema.

Title
cisco-sa-20150923-fhs
cisco-sa-20150325-ikev2
cisco-sa-20160915-ikev1
cisco-sa-20160928-smi
cisco-sa-20160928-ios-ikev1
cisco-sa-20160928-dns
cisco-sa-20160928-ipdr
cisco-sa-20160928-aaados
cisco-sa-20160928-msdp-CVE-2016-6382

Related Links

Tools

Cisco PSIRT openVuln API
Create New TAC Service Request
Query Existing TAC Service Request
Bug Search Tool
Software Downloads
Product Field Notices
End-of-Life Announcements



やってみよう(OVAL定義を見る)

Open Vulnerability and Assessment Language: Element Dictionary

- Schema: IOS Definition
- Version: 5.11.1:1.2
- Release Date: 11/30/2016 09:00:00 AM

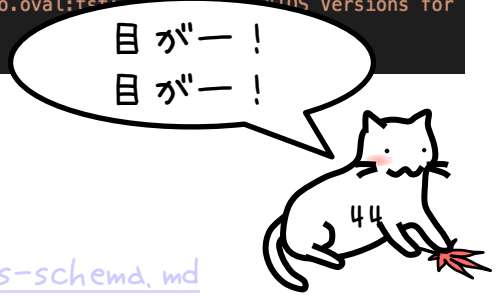
The following is a description of the elements, types, and attributes that compose the IOS sp
Open Vulnerability and Assessment Language (OVAL). Each test is an extension of the standa
in the Core Definition Schema. Through extension, each test inherits a set of elements and at
amongst all OVAL tests. Each test is described in detail and should provide the information n
what each element and attribute represents. This document is intended for developers and a
with XML. A high level description of the interaction between the different tests and their rela
Definition Schema is not outlined here.

The OVAL Schema is maintained by the OVAL Community. For more information, including ho
project and how to submit change requests, please visit the OVAL website at http://oval.cisec

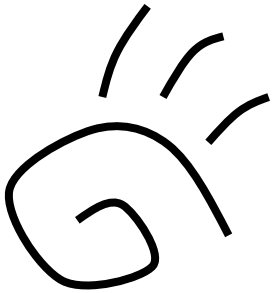
Test Listing

* acl_test
* bgpneighbor_test
* global_test
* interface_test
* line_test
* router_test
* routingprotocolauthintf_test
* section_test
* snmp_test
* snmpcommunity_test
* snmpgroup_test
* snmphost_test
* snmpuser_test
* snmpview_test
* tclsh_test
* version55_test
* ~~version_test~~

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <oval_definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ov
   http://oval.mitre.org/XMLSchema/oval-definitions-5#ios ios-definitions-schema.xs
3    <generator>
4      <oval:product_name>PSIRT OVAL Definition Generator</oval:product_name>
5      <oval:product_version>0.1</oval:product_version>
6      <oval:schema_version>5.10</oval:schema_version>
7      <oval:timestamp>2016-09-27T08:11:49.135088-05:00</oval:timestamp>
8    </generator>
9    <definitions>
10     <definition id="oval:cisco.oval:def:341" version="1" class="vulnerability">
11       <metadata>
12         <title>cisco-sa-20130731-lsaospf-CVE-2013-0149</title>
13         <affected family="ios">
14           <platform>Cisco IOS</platform>
15         </affected>
16         <reference source="CVE" ref_id="CVE-2013-0149" ref_url="" />
17         <reference source="Vendor Advisory" ref_id="cisco-sa-20130801-lsaospf" r
18         <description />
19       </metadata>
20       <criteria operator="AND">
21         <criterion test_ref="oval:cisco.oval:tst:340" comment="router_ospf_check
22         <criterion test_ref="oval:cisco.oval:tst:                "IOS Versions for
23       </criteria>
24     </definition>
```



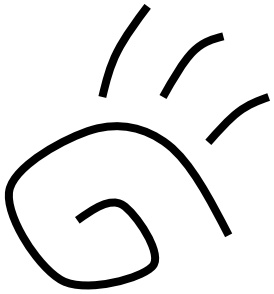
https://github.com/OVALProject/Language/blob/master/docs/ios-definitions-schema.md



やってみよう(OVAL要素)

- generator (発行元)
- definitions
  - metadata (プラットフォーム、関連するCVEなど)
  - criteria (判断の条件)
- tests (テスト内容)
- objects (テストの対象物)
- states (対象物のステータス)

(Ciscoさんが配っているのは、OVAL ver.5.10)



やってみよう(定義の例)

```
20          <criteria operator="AND">
21            <criterion test_ref="oval:cisco.oval:tst:340" comment="router_ospf_check" />
22            <criterion test_ref="oval:cisco.oval:tst:342" comment="IOS Versions for CSCug34485" />
23          </criteria>
```

```
27        <line_test id="oval:cisco.oval:tst:340" version="1" check="all" comment="router_ospf_check" xmlns=
28          <object object_ref="oval:cisco.oval:obj:1" />
29          <state state_ref="oval:cisco.oval:ste:57744" />
30        </line_test>
31        <version55_test id="oval:cisco.oval:tst:342" version="1" check="at least one" state_operator="OR"
32          <object object_ref="oval:org.mitre.oval:obj:6804" />
33          <state state_ref="oval:cisco.oval:ste:4" />
```

```
3652        <line_state id="oval:cisco.oval:ste:57744" version="1" xmlns="http://c
3653          <show_subcommand>show running-config</show_subcommand>
3654          <config_line operation="pattern match">router\s+ospf</config_line>
3655        </line_state>
```

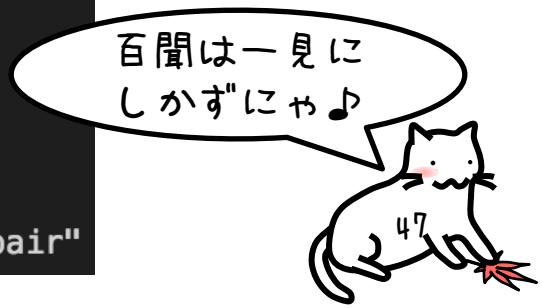
やってみよう(Cisco OVALを見る)

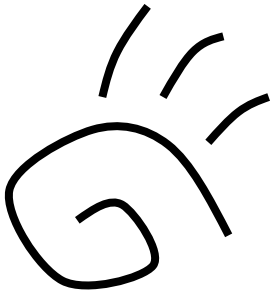
» 実際に使われていたテスト定義

× version55_test (OSバージョンマッチ)

× line_test (設定の文字列マッチ)

```
173 "show_subcommand": "show running-config"
 50 "show_subcommand": "show processes"
 13 "show_subcommand": "show vstack config"
 10 "show_subcommand": "show udp"
  8 "show_subcommand": "show mpls interface"
  8 "show_subcommand": "show ip socket"
  6 "show_subcommand": "show control-plane host open-ports"
  4 "show_subcommand": "show ipv6 dhcp interface"
  4 "show_subcommand": "show ip ips interfaces"
  3 "show_subcommand": "show ip ssh"
  2 "show_subcommand": "show translate"
  2 "show_subcommand": "show subsys"
  2 "show_subcommand": "show process cpu"
  2 "show_subcommand": "show policy-map type inspect zone-pair"
```





やってみた（プログラム書いた）

» プロトタイプ実装

» ネットワークチームから設定サンプルをもらう

× バージョン情報、設定ファイルをスキャン

» できたみたい

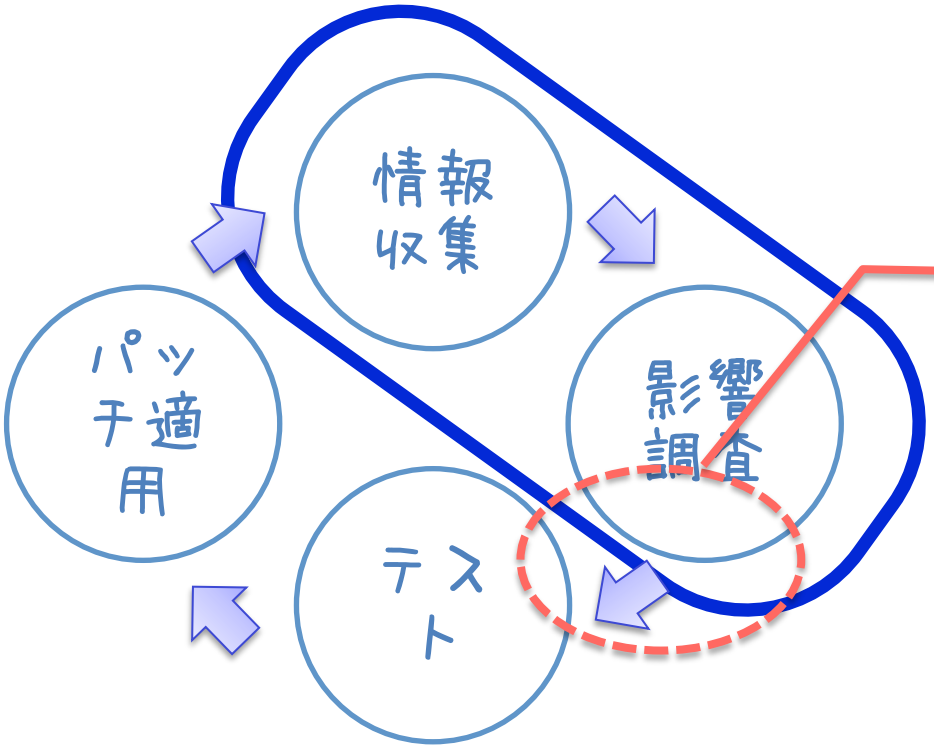
```
checking host sXXXXXXXXgw20
  RESULT: NG    STATE: FOUND router\s+ospf  MATCH 15.2(4)M3
checking host sXXXXXXsw03
  RESULT: -     STATE: MATCH 12.2(25)SEE2
checking host oXXXXXp61
  RESULT: -     STATE: FOUND router\s+ospf
checking host oXXXXXXbb00
  RESULT: NG    STATE: FOUND router\s+ospf  MATCH 12.2(46)SE
checking host oXXXXXXXbi16
  RESULT: -     STATE: MATCH 12.2(33)SRE6
checking host sXXXXXsw01a
  RESULT: -     STATE: MATCH 12.2(25)EWA9
```

気づいたことと、これから

- ネットワーク機器をやってみた
  - × スキャナの分離が必要
    - > 管理主体、管理方法が違うことも多い
  - × 他の手段でスキャンしたものの取り込み
    - > 特殊な環境のスキャナはユーザ自身が実装
- ちゃんとした実装とVulsへの組み込み
- 他のOVALリポジトリへの対応
  - × アドバイザリからのタイムラグは？

検知後の対応も効率化したい

» どのサーバを、どのバージョンに、いつ（優先度）アップデートするか決める必要がある



具体的な対応をする際にも、スピードが求められるため、効率化したい

そこでVulsrepo！

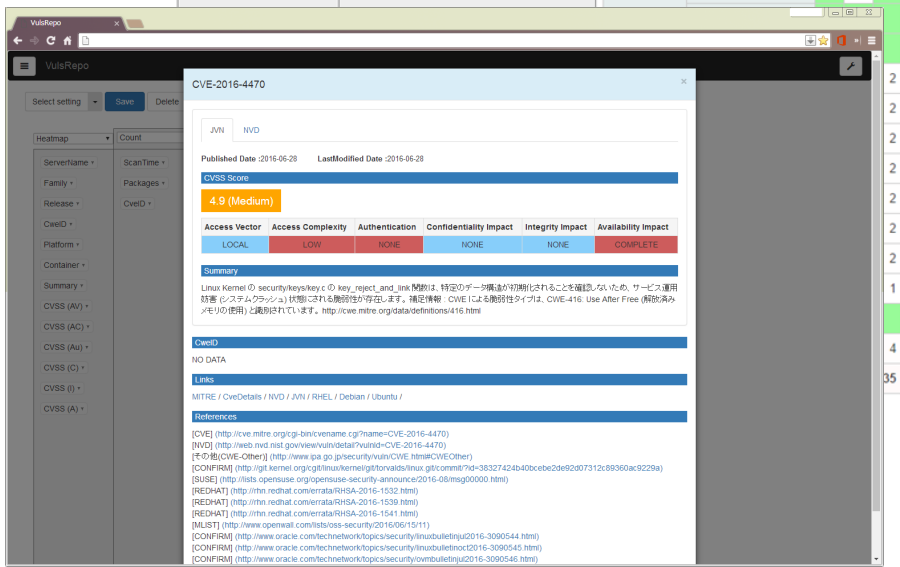
» http://usiusi360.github.io/vulsrepo/

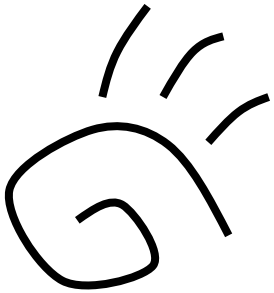
Heatmap | Count | CVSS Severity | CVSS Score

Family | Release | CveID | Packages | Platform | Summary | CVSS (AV) | CVSS (AC)

ServerName | Container

ServerName	Container	CVSS Severity	Low								Medium																		High									Unknown	Totals
		CVSS Score	1.7	1.9	2.1	2.6	2.8	3.3	3.5	3.6	4	4.3	4.4	4.6	4.7	5	5.1	5.5	5.8	6.1	6.2	6.4	6.5	6.8	6.9	7.1	7.2	7.5	7.6	7.8	8.5	9	9.3	10	Unknown	
	centos_5.1			2	6	13		8		4	2	82		8	2	129	20		30	1	5	1	1	46	16		17	75	4	5	1		18	21		517
	centos_5.2			3	10	16		8		6	4	82	2	12	2	159	15		24	1	3	1	2	41	14		13	42	4	4	2		13	20		503
	centos_5.4			1		4		4			2	22		2	1	39	7		4		2	1	1	23	6		8	17	4	1	1		7	7		164
	centos_5.6			2		4		3			1	19		1		31	4		2				1	20				14	4	1	1		7	5		120
	centos_5.7			4		6		1			4	35		1		56	5		6	1				45			3	27	4	2	2		5	6		213
	centos_5.8			2		4					4	29				39	3		6					15			3	23	4	2	2		4	6		146
	centos_5.9			1		2					1	14				14	3		2					5				13	4	1	1			5		66
												13				13	3							4				9	4	1	1			5		53
												12				16	2							3				6	4	6	1			4		54
			2	30						10	216		1		354			17			3		95		22		77		10			2	34	1	885
			2	18						6	132		1		210			13			3		71		14		65		6			2	22	1	573
			2	18						6	117		1		173			13			3		63		14		57		6			2	22	1	505
			2	18						6	118		1		173			13			3		60		14		62		6				22	1	506
			2	4						4	81				122			9			3		47		10		43		4				16	1	350
			2	1							25				53			5			3		30		4		26		1				7	1	159
			2	1							23				44			3			3		20		3	1	17		7				5	1	131
			1	1							11				23			3					18		1	1	15		7				5		87
				2											10			2					16						2				8		40
			4	3	1		6			13	11		2		27		3	2				3	19	2	1	14	15		2		1		8	3	144
			35	145	1	24	6	10		63	1,042	2	30	5	1,685	62	3	154	3	10	24	8	641	38	83	60	603	36	74	12	1	60	228	10	5,216





レポートを求められることも多い

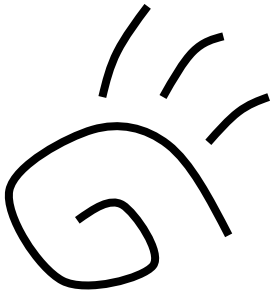
» セキュリティインシデント発生による会社へのインパクトが大きい
→ 経営者、セキュリティ責任者（CISO）

他にも多くのレポート手段

- Kibana
  http://54.238.135.81:5601/app/kibana#/dashboard/Vuls-Dashborad
- Excel
  https://1drv.ms/x/s!AqN7HjAnUKpigTDIiuKY07LtfT_5
- QuickSight
  https://us-east-1.quicksight.aws.amazon.com/sn/dashboards/25ddd4c2-fe80-484f-b0c2-5e41586447f4

第三者への対応も必要

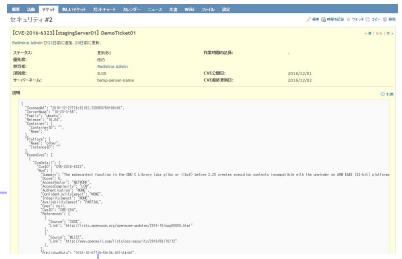
» 内部や利害関係者への報告だけでなく、
対外的に脆弱性の管理をしていることを示す
必要がある。
→　外部監査
→　コンプライアンス、ガイドライン対応

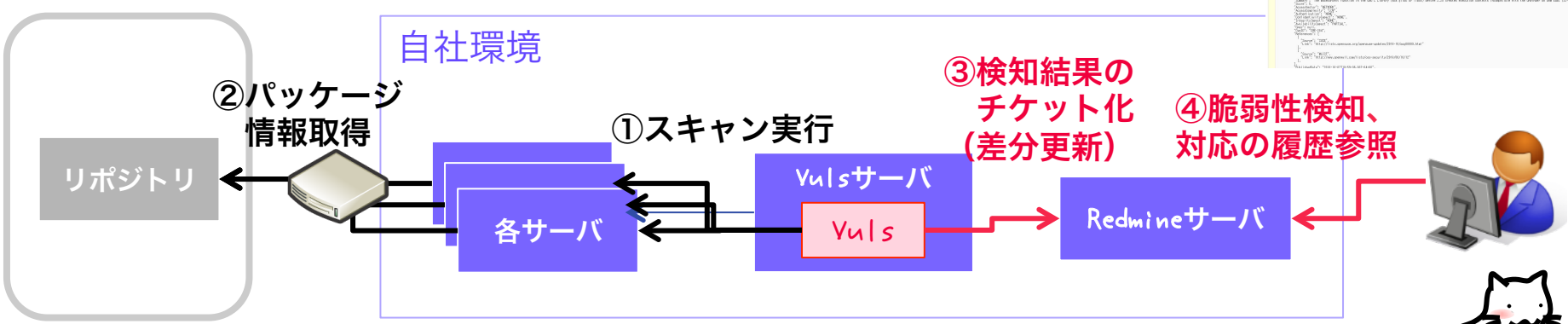


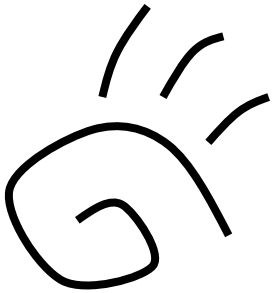
Redmineと連携して履歴管理

» どのサーバのどの脆弱性にいつ、どんな対応をしたかの履歴が分かる。

» セキュリティは継続することが必須。運用者が変わっても対応しやすい。







Zabbixと連携も可能

》現行環境との親和性も高い

脆弱性検知結果を出力・加工

```
$ vuls scan -report-json
$ vuls scan -report-mail
$ vuls scan -report-s3
$ vuls scan -report-slack
$ vuls scan -report-text
```

様々な出力形式に対応

```
# 脆弱性の数
$ cat vuls/results/current/target1.json | jq
'[.KnownCves[]?, .UnknownCves[]?
| .CveDetail.CveID] | length'
```

79 ← 脆弱性の数を取得

```
# Cvss Scoreの最大値
$ cat vuls/results/current/target1.json | jq
'[.KnownCves[]?, .UnknownCves[]?
| .CveDetail.Nvd.Score] | max'
```

10 ← CVSSスコアを取得

ZABBIX

スキャンスクリプトをcronに登録し、Zabbixに送信

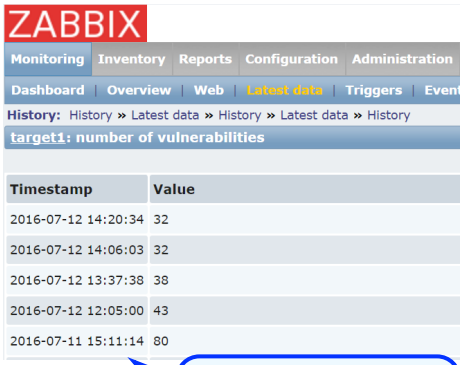
```
#!/bin/bash
VULS_ROOT="/root/vuls"
VULS_RESULT_DIR="$VULS_ROOT/results"

vuls scan -cve-dictionary-dbpath=$VULS_ROOT/
cve.sqlite3 -report-json
files="$VULS_RESULT_DIR/current/*"
for filepath in $files; do
    TARGET_NAME=`basename $filepath .json`
    zabbix_sender -z localhost -s $TARGET_NAME -k
nvd_count -o `cat $filepath | jq
'[.KnownCves[]?, .UnknownCves[]? | .CveDetail.CveID] |
length'`
    zabbix_sender -z localhost -s $TARGET_NAME -k
nvd_max -o `cat $filepath | jq
'[.KnownCves[]?, .UnknownCves[]?
| .CveDetail.Nvd.Score] | max'`
done
```

zabbix senderを使って脆弱性情報を送信するスクリプトを定期実行

ZABBIX

Zabbixで条件指定、新規脆弱性検知



条件指定、新規脆弱性検知、などをZabbixで組み込む

他ソフトウェアとの連携

» 今後も連携機能増えるかも？

× FARADAY by Infobyte LLC

× REDMINE flexible project management

× ZABBIX

⋮

議論したいところ

- みなさん、どうしてますか？
- これって使えそうだと思いましたか？
- どんなツールで運用を実践してますか？
- 他にいいツールを知りませんか？

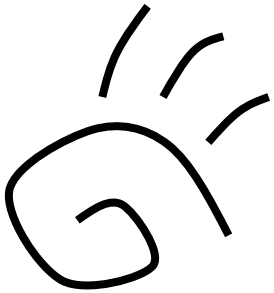
みなさん、どうしてますか？

» 運用はバッチリ！パッチが出たら即適用！

» 気合いでしのぎつつ、タイミングを見て

» 緩和策命

» ノーガード



これって使えそうだと思いましたか？

» Yeah!

» Booooooo!

どんなツールで運用を実践してますか？

» 我こそは！

» 限定的にでも！一部人力でも！

他にいいツールを知りませんか？

» こんなのつなげたい！

» こうやったら、あれもできるよ！

» もっとこうして欲しい！

以上！

ご静聴ありがとうございました
俺たちの戦いはまだ続くぜ！