# Agenda

■CVSSミニチュートリアル

■Exploitablity(攻撃される可能性)をどう考えるか?

■BoF(バッファオーバーフロー)脆弱性は直ちに危険か?

■MITM前提脆弱性への対応温度感

■Internet Unreachableな脆弱性は放置できるか?

■セキュリティ原理主義とどう向き合うか～情報共有のすゝめ～

# 運用者がCVSSを活用するべき3つの理由

## ①定量化による一次判断の効率化・迅速化

- 専門家の知見に基づいて作られたフレームワーク、独自手法に勝る
- セキュリティの専門知識を有さずともある程度の判断が可能

## ②運用者目線で対応優先度を下げるロジックとして有用

- 仕組みを理解し、条件を設定すればScore(深刻度/優先度の評価値)は低下、低下しない脆弱性は本当に要対応な脆弱性

## ③見えないものが見えてくる(かも)

- CVSS評価値は各種バイアスの影響を受けにくい
- Vectorからメーカーの想いが滲みることがある
  (情報の非対称性の垣根を超えたヒントが含まれることがある)

# CVSS：共通脆弱性評価システム

Common Vulunerability Scoring System：脆弱性の定性的な評価を組み合わせ、計算式により深刻度を定量化する仕組み

**CVSS Score:** 7.8

**CVSS Vector:** (AV:N/AC:L/Au:N/C:N/I:N/A:C)

■一般用途：対応基準の明確化による平準化、水準の底上げ

- ●組織規定
  例)顧客向けサービススペック、社内情報セキュリティ規定
- ●業界基準
  例)PCIDSS:Payment Card Industry Data Security Standards
  - ・インターネット公開システム：CVSS Base Score 4.0以上の脆弱性がないこと
  - ・内部システム：高リスク脆弱性(Severity:High=Score:7以上)が解消されること

CVSSミニチュートリアル

# ①定量化による一次判断の効率化・迅速化

■数多ある脆弱性情報の対応要否、優先度や**概要**を迅速に判断

**CVSS Score:** 7.8

**CVSS Vector:** (AV:N/AC:L/Au:N/C:N/I:N/A:C)

[共通脆弱性評価システムCVSS概説]
https://www.ipa.go.jp/security/vuln/CVSS.html
[共通脆弱性評価システムCVSS v3概説]
https://www.ipa.go.jp/security/vuln/CVSSv3.html

## Score=深刻度

| CVSS Score | Severity |
|---|---|
| 9.0～10.0 | Critical |
| 7.0～8.9 | High |
| 4.0～6.9 | Middle/Medium |
| 0～3.9 | Low |

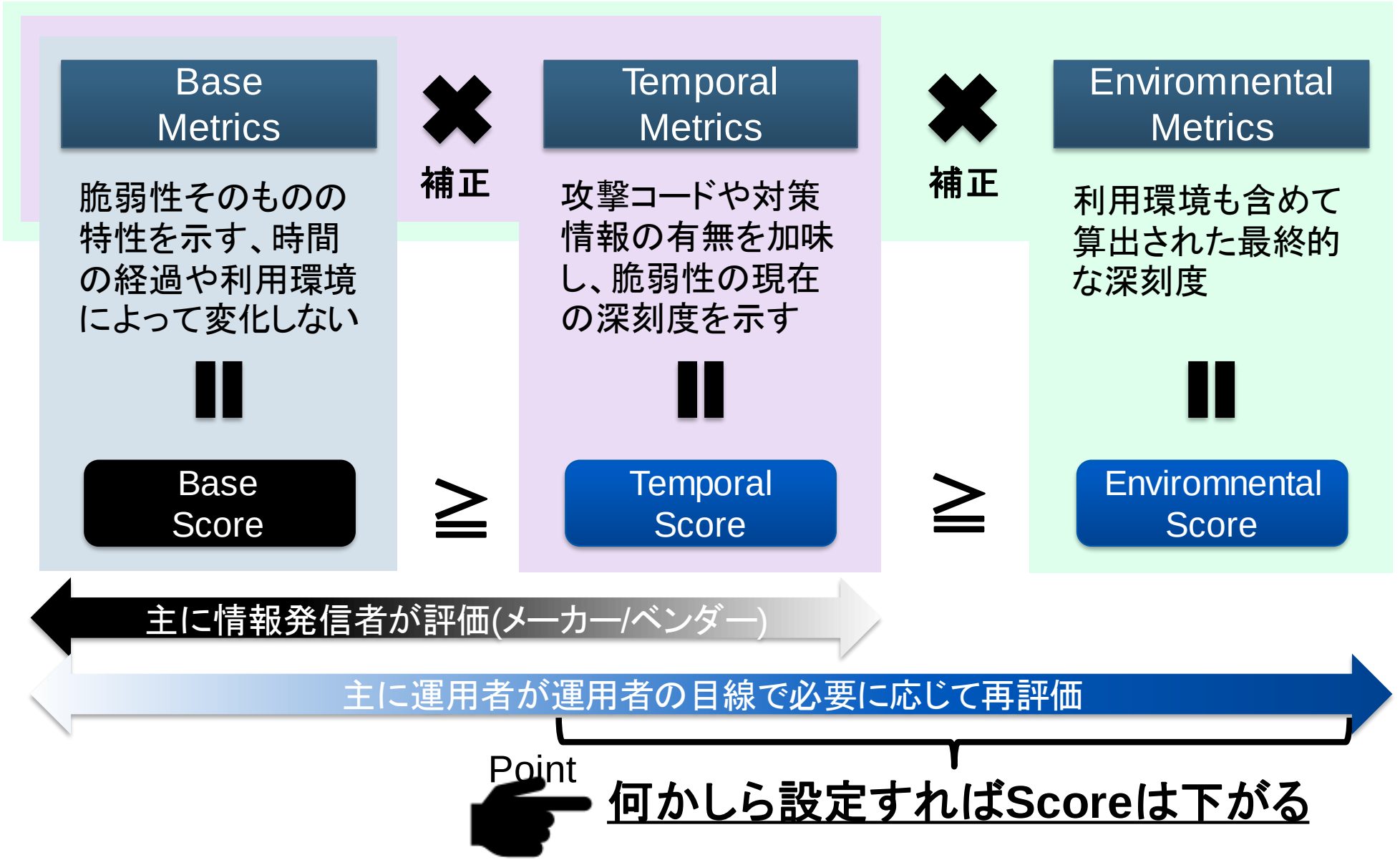
## Vector=概要とScoreの根拠

| | |
|---|---|
| AV(攻撃元区分) | N(ネットワーク経由) |
| AC(攻撃の複雑さ) | L(低) |
| Au(攻撃前の認証要否) | N(不要) |
| C(機密性への影響) | N(なし) |
| I(完全性への影響) | N(なし) |
| A(可用性への影響) | C(完全) |

脆弱性情報(Security Advisory)にはCVSS評価の詳細明記が通例

CVSSミニチュートリアル

# ②運用者目線で対応優先度を下げるロジックとして有用



# ③見えないものが見えてくる（かも）

## ■情報の質と量の非対称性

運用者 << 発信者 (メーカー/ベンダー)

CVSSが各種バイアスを排除したほぼ唯一の共通基準、共通言語

## ■CVSSから滲み出てくる発信者の想い(?)

●Case1)説明は”Unspecified Vulnerability”なのにVectorは細かく明記

例) Base AV:N/AC:L/Au:N/C:N/I:N/A:C
Temporal E:POC/RL:OF/RC:C
Environmental CDP:ND/TD:H/CR:ND/IR:ND/AR:ND

最悪でもサービス停止なのね

●Case2)CVSS ScoreとSeverityの乖離

- CVSS Score < Severity
  ➔やばそう
- CVSS Score > Severity
  ➔たいしたことなさそう

| CVSS Score | Severity |
|---|---|
| 9.0～10.0 | Critical |
| 7.0～8.9 | High |
| 4.0～6.9 | Middle/Medium |
| 0～3.9 | Low |

# CVSSのバージョン

## ■歴史

- ●バージョン1：2005年2月リリース
- ●バージョン2：2007年6月リリース
- ●バージョン3：2015年6月リリース

[参考]
共通脆弱性評価システムCVSS概説
https://www.ipa.go.jp/security/vuln/CVSS.html
共通脆弱性評価システムCVSS v3概説
https://www.ipa.go.jp/security/vuln/CVSSv3.html

## ■現在

- ●現在はv2を基本に、v3を併記することが徐々に増えている、v1は使われていない
  単にCVSSと書かれている場合はv2を示すと思えばよい
- ●v3はv2の評価基準を見直し、さらに細分化、再整理したもの、より脅威の実態に即したに評価ができるようになった一方、少し複雑化している
- ●どこかでv3へ完全移行のタイミングが来るかもだけどよくわらない、現在過渡期
- ●**運用者としてはCalicuratorの選択パラメータの変化くらいしかないから影響軽微**

3つのMetricsとCaliculatorの使い方を覚えればOK

CVSSミニチュートリアル

# Caliculatorに触ってみよう

例)http://bit.ly/2j6ouWK

**National Cyber Awareness System**

**Vulnerability Summary for CVE-2016-6366**

**Original release date:** 08/18/2016

**Last revised:** 11/28/2016

**Source:** US-CERT/NIST

## Modified

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

## Overview

Buffer overflow in Cisco Adaptive Security Appliance (ASA) Software through 9.4.2.3 on ASA 5500, ASA 5500-X, ASA S
ASA Security Module, PIX, and FWSM devices allows remote authenticated users to execute arbitrary code via crafted I
EXTRABACON.

## Impact

**CVSS Severity (version 3.0):**

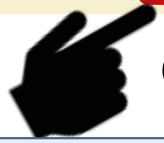
**CVSS v3 Base Score:** 8.8 High

**Vector:** CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

**CVSS Severity (version 2.0):**

**CVSS v2 Base Score:** 8.5 HIGH

**Vector:** (AV:N/AC:M/Au:S/C:C/I:C/A:C) (legend)



Click

脆弱性情報(Security Advisory)にCaliculatorへのリンクが存在することが多い

CVSSミニチュートリアル

# 読みといてみる、計算してみる

| | |
|---|---|
| **CVSS Base Score** | **8.5** |
| Impact Subscore | 10 |
| Exploitability Subscore | 6.8 |
| **CVSS Temporal Score** | **Not Defined** |
| **CVSS Environmental Score** | **Not Defined** |
| Modified Impact Subscore | 0 |
| **Overall CVSS Score** | **8.5** |

Show Equations

**CVSS v2 Vector** (AV:N/AC:M/Au:S/C:C/I:C/A:C)

**Base Score Metrics**

**Exploitability Metrics**

Access Vector (AV)*

Local (AV:L) | Adjacent Network (AV:A) | **Network (AV:N)**

Access Complexity (AC)*

High (AC:H) | **Medium (AC:M)** | Low (AC:L)

Authentication (Au)*

Multiple (Au:M) | **Single (Au:S)** | None (Au:N)

**Impact Metrics**

Confidentiality Impact (C)*

None (C:N) | Partial (C:P) | **Complete (C:C)**

Integrity Impact (I)*

None (I:N) | Partial (I:P) | **Complete (I:C)**

Availability Impact (A)*

None (A:N) | Partial (A:P) | **Complete (A:C)**

* - All base metrics are required to generate a base score.

**Temporal Score Metrics**

**Environmental Score Metrics**

**[共通脆弱性評価システムCVSS概説]**
https://www.ipa.go.jp/security/vuln/CVSS.html
**[共通脆弱性評価システムCVSS v3概説]**
https://www.ipa.go.jp/security/vuln/CVSSv3.html

# CVSSまとめ

## ■専門家の知見を誰でも使えるフレームワークに落としたもの

- ●あれこれ考える必要がない、車輪の再発明はするべきでない
- ●誰でも使えるようにシンプルにまとめられている、評価結果はオープン
- ●一度理解するとすごく楽になる、ハンドリングを効率化できる

## ■使いこなせば簡単便利なツール

- ●メーカーやベンダーの評価値 x 利用者の再評価で総合評価
- ●実際の作業はCaliculatorでポチポチ選択するだけ

## ■メーカーの想いが垣間見える

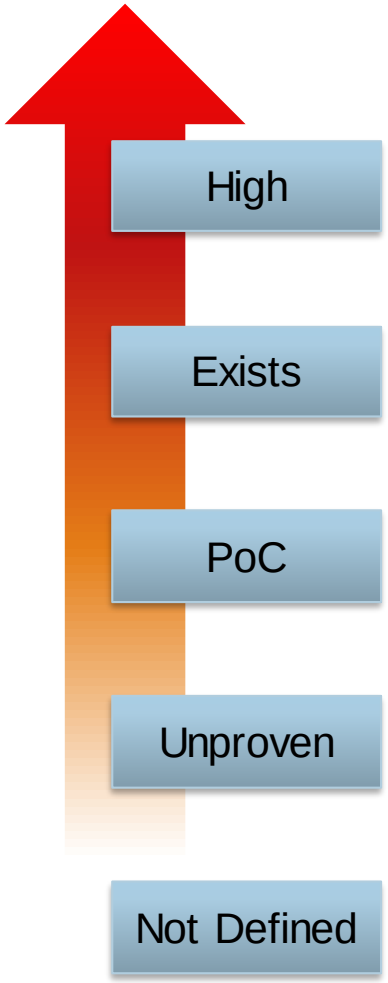
- ●各種バイアスを排除したほぼ唯一の共通基準、共通言語
- ●ScoreとSeverityの不一致には意味がある

# 議論

■CVSSってご存じでしたか?使いこなしてますか?課題ありますか?

## Exploitablityをどう考えるか？

# Exploitability：攻撃される可能性



| | | |
|---|---|---|
| High | 容易に攻撃可能 | 攻撃コードがいかなる状況でも利用可能である<br>攻撃コードを必要とせず、攻撃可能である |
| Exists | 攻撃可能 | 攻撃コードが存在し、ほとんどの状況で使用可能である |
| PoC | 実証可能 | 実証コードが存在している<br>完成度の低い攻撃コードが存在している |
| Unproven | 未実証 | 実証コードや攻撃コードが利用可能でない<br>攻撃手法が理論上のみで存在している |
| Not Defined | 未評価 | この項目を評価しない |

[参考]共通脆弱性評価システムCVSS概説
https://www.ipa.go.jp/security/vuln/CVSS.html

Exploitablityをどう考えるか？

# Exploit作成容易性を計る尺度

## ■オープンソースとクローズドソース

- ●オープンソース)修正差分(diff)をチェックすることで比較的推測が容易
- ●クローズドソース)詳細がつまびらかにならないため一般に推測が困難

## ■Acknowledgement(謝辞)の有無

- ●有)発見者のリーク、あるいは示唆により推測される可能性あり
- ●無)自主的に改修されたものであり、詳細が市中に出回る可能性は低い

実際、注視している組織や担当者は多い

# 加速する情報公開（最近のExploit公開場所）

## ■Pastebin

- ●誰でも匿名でテキストデータを保存し公開することができるWebサービス
- ●その多くが掃き溜めサイトと化しており、PoCやExploitコードも多い
- ●pastebin.comはその代表
  *CVE-2015-5477（BIND TKEY脆弱性）のExploitはここで公開*

## ■GitHub

- ●言わずと知れたオープンソース開発プラットフォーム、PoCやExploitも多い
- ●ここに誰かがリポジトリを作成したりPull Requestした時点でつまびらかになる
- ●Metasploit Framework(ペネトレーションテストツール)のリポジトリは要注意
  https://github.com/rapid7/metasploit-framework

  *CVE-2016-2776(BIND TSIG脆弱性)のExploitはここで公開*

## Exploitablityをどう考えるか？

# 加速する情報公開（昔ながらのExploit公開場所）

## ■攻撃コードデータベースサイト

- ●攻撃コードを集めた専門サイト
- ●建前(?)はペネトレーションテスト利用を目的とする
- ●exploit-db.comはその代表


## ■脆弱性情報データベースサイト

- ●あらゆる製品の脆弱性情報を集めた情報共有サイト
- ●その一環で攻撃コードも公開
- ●SecurityFocusはその代表

Exploitablityをどう考えるか？

# 議論

バッファオーバーフロー脆弱性は直ちに危険か？

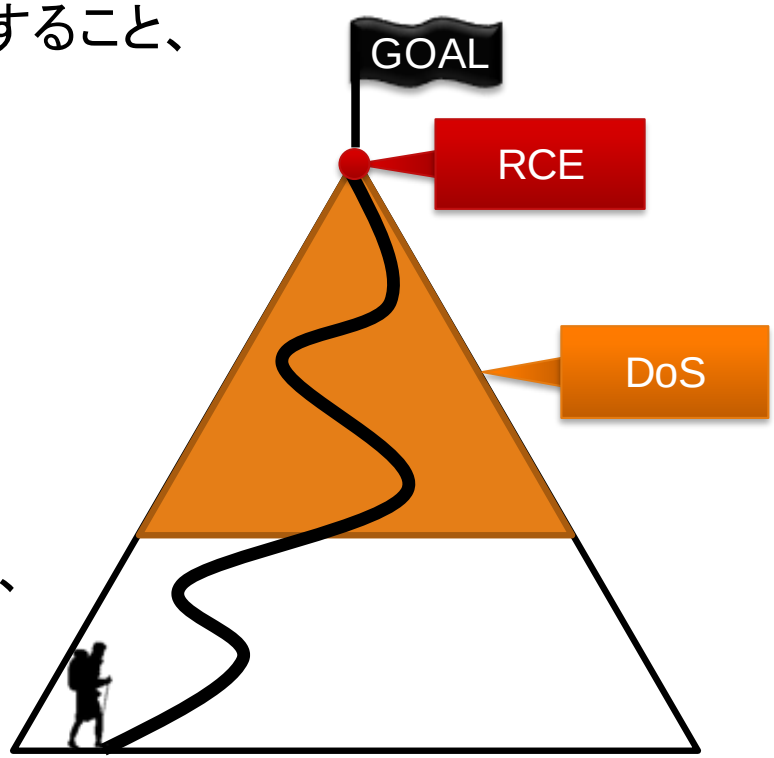
# BoF (Buffer Overflow) 脆弱性とRCE、DoS

## ■RCE(Remote Code Execution：リモートからの任意のコード実行)

- ●メモリ破壊によりリモートから任意のコードを実行すること、それを引き起こす脆弱性
- ●機器を乗っ取ってなんでもできる、悪意のある攻撃者はここを目指す

## ■DoS(Denial of Service：サービス妨害)

- ●メモリを破壊しているものの何らかの理由によりRCEまでは至らずプログラムが異常停止すること、それを引き起こす脆弱性



Mt. Buffer Overrun

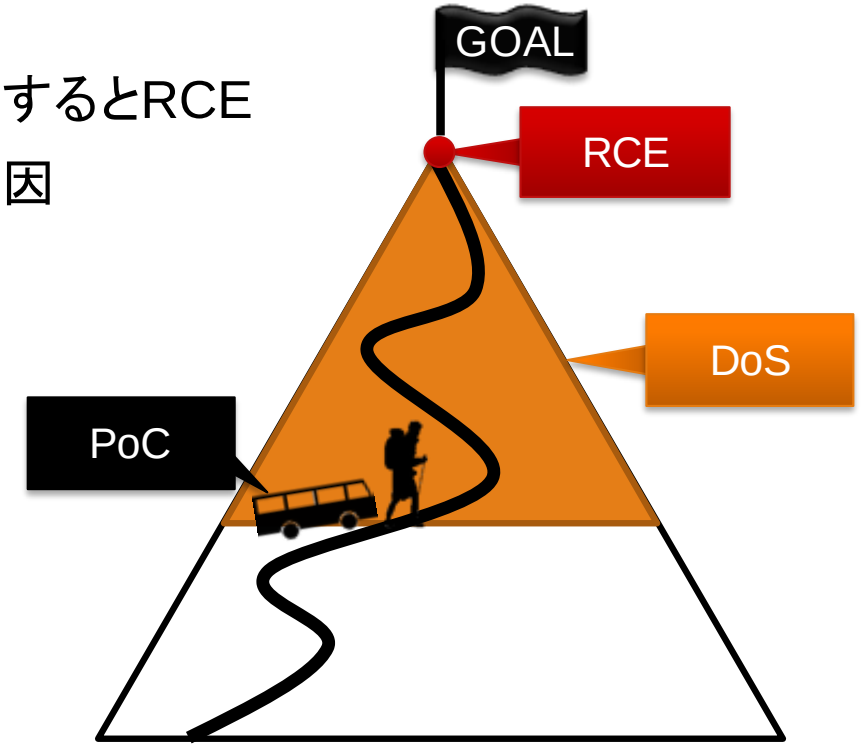
リモートから攻撃可能なBoF脆弱性は無条件にSeverity:High/Critical

バッファオーバーフロー脆弱性は直ちに危険か？

# BoF脆弱性事例①：glibc getaddrinfo () RCE脆弱性

## ■概要

- CVE-2015-7547
- 利用者が悪意のあるDNSレコードを名前解決するとRCE
- glibcに含まれるgetaddrinfo()の実装ミスが原因
- Google社/Redhat社が共同で報告



Mt. Buffer Overrun

## ■攻撃難易度

- Google社からDoSに至るPoCが公開済
- 5合目から頂上を目指すようなもの
- メモリ保護機能を有したモダンなOSでは難易度が高い

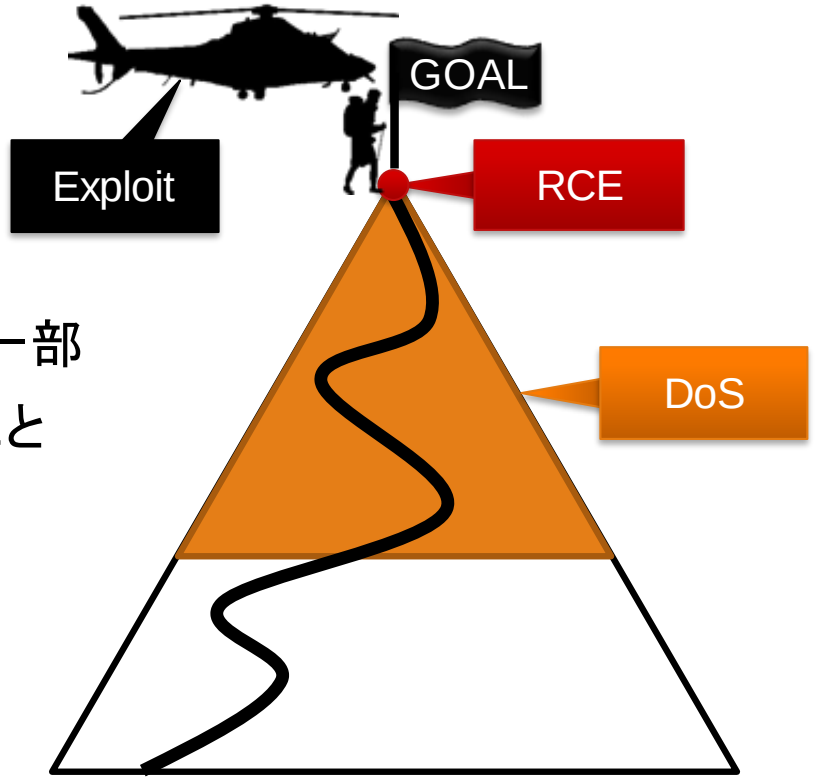
**一方、ネットワーク機器は？流行りのIoTデバイスは？**

バッファオーバーフロー脆弱性は直ちに危険か？

# BoF脆弱性事例②：Cisco Firewall RCE脆弱性（EXTRA BACON）

## ■概要

- CVE-2016-6366
- Cisco ASA/PIXのsnmp実装のBoFが原因
- 攻撃者集団Shadow Brokerが、
  攻撃者集団Equation Groupから盗み出した
  非公開の攻撃コードを小出しに公開したものの一部
- snmpデーモンをインターネットに公開していること
  はまずないが、昨今のマルウェアを悪用する
  標的型攻撃を想定すると放置できない脆弱性




## ■攻撃難易度

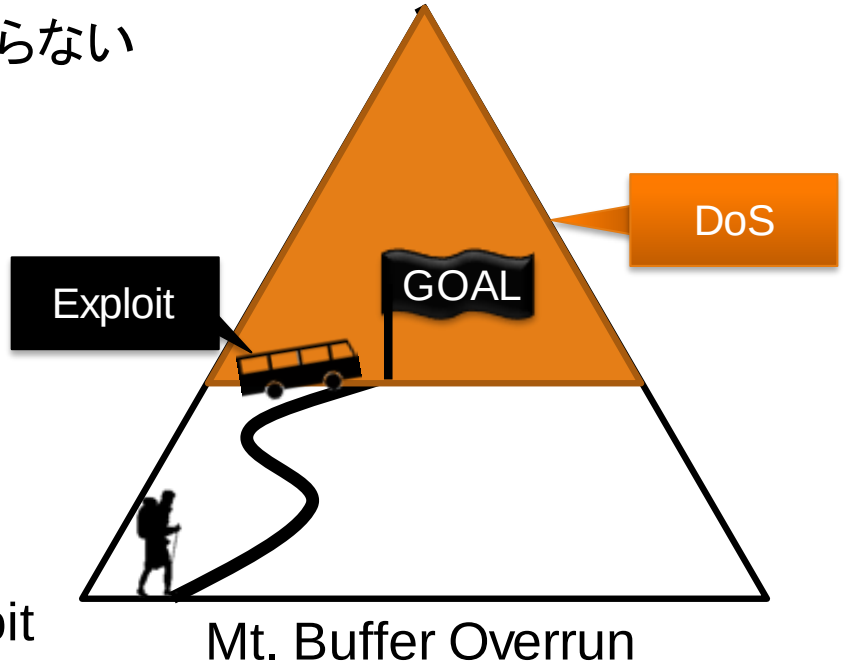
- 大手攻撃コードサイトでExploitが公開済
- 誰でもヘリコプター(Exploit)で登頂(RCE)可能な状態

バッファオーバーフロー脆弱性は直ちに危険か？

# BoF脆弱性事例③：BIND DoS脆弱性

## ■概要

- CVE-2016-2776/CVE-2015-5477等
- BINDの実装不備により、パケット一発でプロセス停止
- BINDの設計思想により、BoFでもRCEには至らない
- Workaroundなし、実際に国内でも被害を観測




## ■攻撃難易度

- DNSにとってはDoSだけでも大脅威
- そもそものGOALが低い
  山の五合目まで到達すればよい、PoC=Exploit
- バス(Exploit)がインターネット上で公開された
  #opkillingxxxの真っ只中、悪用されていたら・・・？

バッファオーバーフロー脆弱性は直ちに危険か？

# BoF脆弱性悪用によるRCEの難易度

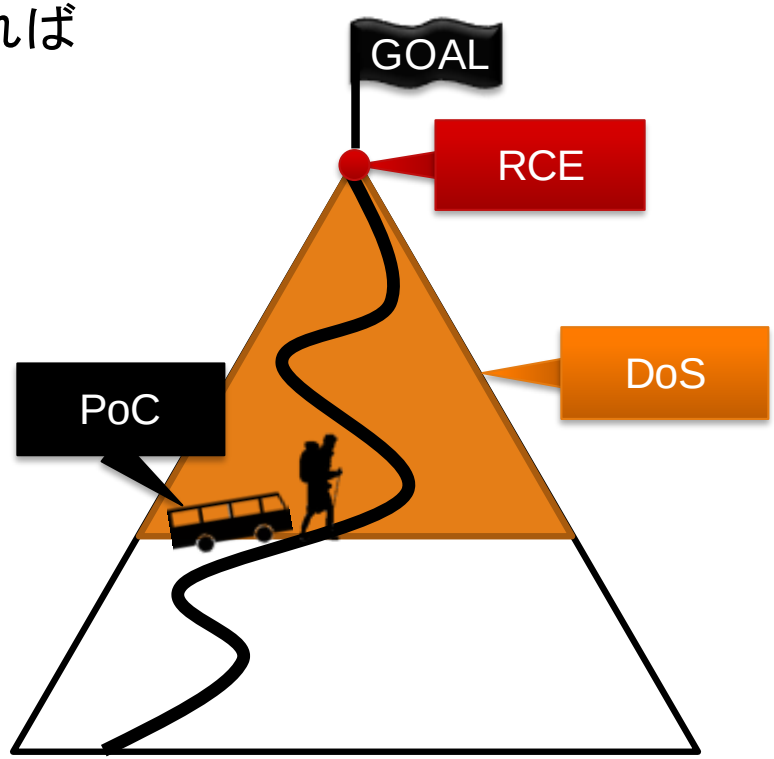
## ■昔

- 大学でコンピュータサイエンスを嗜んだ学生であれば攻撃および攻撃コードの作成が可能

## ■現在

- OSの保護機能によりそんなに簡単じゃない

| | |
|---|---|
| メモリ保護 | Stack Smash Protection<br>RELRO(RELocation ReadOnly) |
| メモリランダマイズ | ASLR(Address Space Layout Randomization)<br>PIE(Position Independent Executable) |
| データ実行防止 | NXbit(No eXecute bit)<br>DEP(Data Execution Prevention) |
| その他 | Control Flow Guard |




100%安全とまでは言えないが、攻撃および攻撃コードの作成難易度は高い

## バッファオーバーフロー脆弱性は直ちに危険か？

# 統計データ

■(その場限り)

バッファオーバーフロー脆弱性は直ちに危険か？

# 議論

■(その場限り)

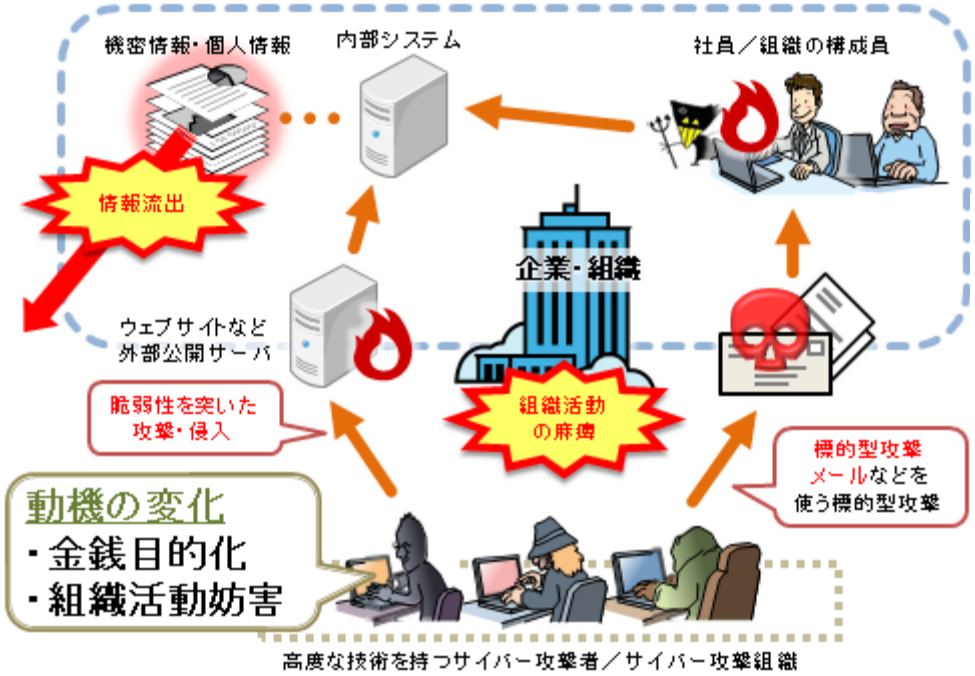
MITM前提脆弱性への対応温度感

# 議論

■(その場限り)

## Internet Unreachableな脆弱性は放置できるか?

# 標的型攻撃とEXTRA BACON(CVE-2016-6366)系脆弱性を考える



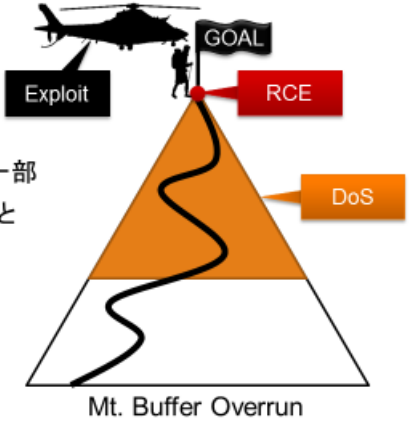

[引用元]
http://www.ipa.go.jp/security/txt/2012/01outline.html

バッファオーバーフロー脆弱性は直ちに危険か？

### BoF脆弱性事例②:Cisco Firewall RCE脆弱性 (EXTRA BACON)

■概要

- ●CVE-2016-6366
- ●Cisco ASA/PIXのsnmp実装のRCE脆弱性
- ●攻撃者集団Shadow Brokerが、攻撃者集団Equation Groupから盗み出した非公開の攻撃コードを小出しに公開したものの一部
- ●snmpデーモンをインターネットに公開していることはまずないが、昨今のマルウェアを悪用する標的型攻撃を想定すると放置できない脆弱性




■攻撃難易度

- ●大手攻撃コードサイトでExploitが公開済
- ●誰でもヘリコプター(Exploit)で登頂(RCE)可能な状態

すぐ対応しないにせよ、いつまでも放置はまずくないですか?

## Internet Unreachableな脆弱性は放置できるか?

# 議論

■EXTRA BACON (CVE-2016-6366)対応しました?

■どのくらいの期間で対応しました？

**セキュリティ原理主義とどう向き合うか～情報共有のすゝめ～**

■(別資料、その場限り)