STRIDEから始めた脅威モデルの進化

コンフェルダー・ローレン

自己紹介 Introduction

MCT

TAITO®

Lawrence Livermore National Laboratory

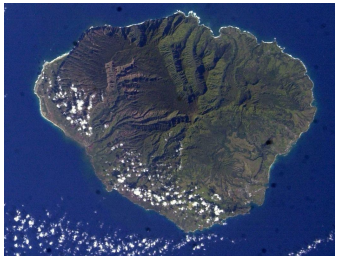
handon software inc.

ELXSI

Canon

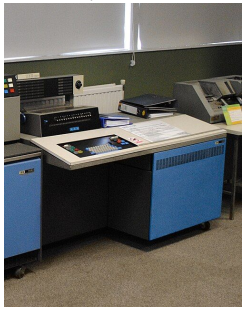
Microsoft

Google



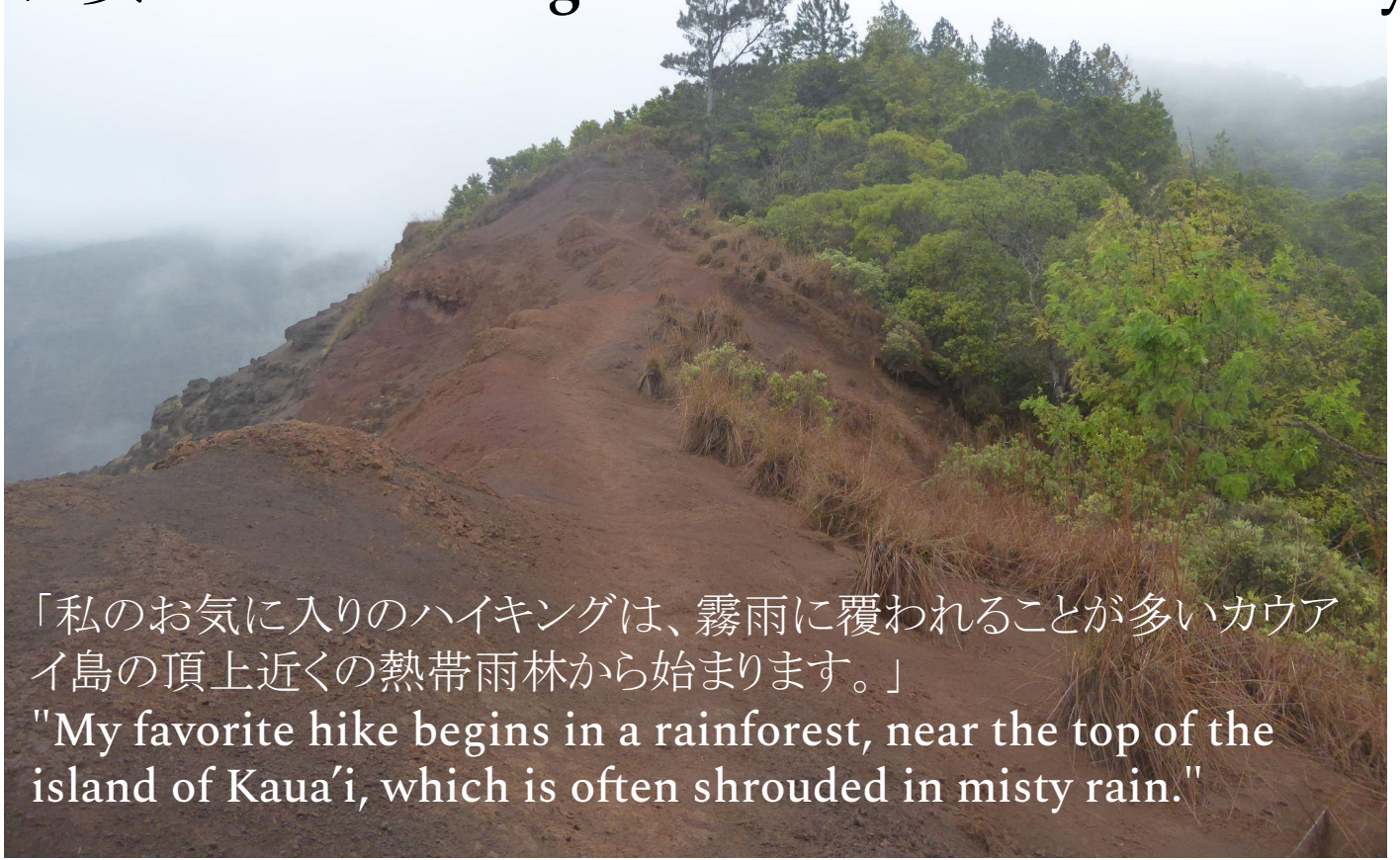
1970年代 → 1980年代 → 1990年代 → 2000年代 → 2010年代 → 2020年代

- Digital certificates
- IE3&4 security
- Security/Privacy teams
- NeXTSTEP日本語化
- STRIDE
- .NET security



Mainframe, Minicomputer, PC, GUI, internet, WWW, Mobile, Cloud, Machine Learning

初めに安心　Let's begin a safe software security path



「私のお気に入りのハイキングは、霧雨に覆われることが多いカウアイ島の頂上近くの熱帯雨林から始まります。」
"My favorite hike begins in a rainforest, near the top of the island of Kaua′i, which is often shrouded in misty rain."

過去と現在 Then and Now

ソフトウエア57年、日本10年の経験

セキュアなソフトウェア (本)

⇒AIの転換期

⇒ソフトウェアへの文化の影響

⇒セキュリティの停滞は懸念される

「脅威モデルは、セキュリティ特性を指定し、設計が何を予測しているか、そしてどのように保護が提供されているかを説明するのに最適な方法です。」

Software 57 years (10 in Japan)

Designing Secure Software (book)

⇒AI at a turning point

⇒Cultural impact on software

⇒Security stagnation is worrisome

"Threat models are the best methodology we have to specify the security properties of technology, explain what the design anticipates, and how protection is provided."

ソフトウエアの進化 Software evolution

旧式のソフトウェア：シンプルなシステム、個別のアプリケーション
Old school software: thin system, separate applications

最新のアプリ：ネットワークインターフェース、深いスタック依存関係
Modern apps: network interfaces, deep stack dependencies

ウエッブ・アプリ：最新のブラウザの複雑さ、Web API、フレームワーク
Web apps: modern browser complexity, web API, frameworks

相互作用と依存関係 → 共通の脅威モデル
Interactions and dependencies → shared threat models

セキュリティとは？　What is "security"?

セキュリティの定義？メトリック？

後付けは難しい／できない

トレード・オフ：機能 対 品質

要件、反要件（継続的に要調整）

「セキュリティ専門家」はなぜ?

簡単な事こそ、不可解

→ 本質的な修復不可能な欠陥
または根深い誤解？

Security: Definition? Metrics?

Build it in, hard/impossible to add-on later

Functionality vs Quality tradeoff

Requirements & anti-requirements (timely)

Why do we have "security experts"?

Even simple aspects seem unknowable

→ Inherent unfixable flaw
or deep misconceptions?

セキュリティのあり方　The state of software security

すべてのソフトウェア（単純なコードを除く）にはバグがあります。
All software (except for simple code) has bugs.

ホフスタッターの法則: この法則を考慮しても、開発には予想以上に時間がかかります。
Hofstadter's law: Development takes longer than expected, even adjusting for this law.

セキュリティは目的地ではなく、方向性。
Security is journey, not a destination.

セキュリティ問題は山ほどあって、結構です。何の代替案は提案されていますか?
Security problems abound: this is fine. What alternative is even being proposed?

脅威モデルとは？ What is threat modeling?

方法論よりも概念 (conceptual over methodology) ＜最も抽象的、万能＞

Shostack's Four Question Framework

1. 何に取り組んでいるのか？ What are we working on?
2. 何が問題を引き起こす可能性があるか？ What can go wrong?
3. それに対して何をするのか？ What are we going to do about it?
4. 充分に良い対応をしたか？ Did we do a good job?

脅威のモデル化はセキュリティにとって不可欠：脅威の認識が保護に先行します。
Threat modeling is essential for security: threats awareness precedes protection.

関係者の皆さんでも、簡単に4質問に答えることができるよいに目指しましょう。
Everyone involved having basic answers for these 4 questions is a great goal.

STRIDE (1999)

脅威のカテゴリーのリスト＜抽象的＞
(abstract categories for InfoSec threats)

脅威のカテゴリー **Threat category**	セキュリティ属性 **InfoSec attribute**
Spoofing なりすまし	Authenticity 真正性
Tampering 改ざん	Integrity 完全性
Repudiation 否認	Non-repudiability 否認不能性
Information disclosure 情報漏洩	Confidentiality 機密性
Denial of service サービス拒否	Availability 可用性
Elevation of privilege 権限昇格	Authorization 承認

反論と言い訳　Objections and excuses...

知らないから（ただの流行だと思っている、脅威インテリジェンスと勘違い）、

必要ないから（日本ではオフェンシブセキュリティが全盛である）、

できないと思っているから（プロセスは知らない）、

時間が無いから。

<u>対策のアイデア</u>

- 組織や要求に合わせて自由にカスタマイズしてもよい
- 小さく始めればよい
- 認証基準ではなくセキュリティ向上のためである
- 網羅できなくても次の回に活かせればよい

日本特有の要因 Japanese cultural factors

脅威モデルの実施に影響を及ぶ課題の一部をリストアップ　List (partial) of potential influences

1. 完璧主義と失敗回避 (改善)　Perfectionism and failure avoidance
2. 集団調和と衝突回避　Group harmony and conflict avoidance
3. 階層と年功序列（先輩後輩）Hierarchy and seniority
4. 長期雇用と暗黙知の伝達　Long-term employment and tacit knowledge transmission
5. プロセス指向の管理（経営）と既存の慣行　Process-oriented management and precedent
6. 「頑張り」と努力による困難克服　Overcoming difficulties through hard work and effort
7. 顧客中心主義とおもてなし　Customer-centricity and hospitality
8. リスク回避と実績のある方法への嗜好　Risk avoidance and proven methods preference
9. 言葉の曖昧、謙遜　Vague language vs explicit expression, humility
10. お任せ　Defer to experts vs "My Way"

（これ以上、沢山あるでしょう）... and doubtless many more

脅威モデルの将来的な進化
Future evolution of threat modeling

脅威モデルは必須と認識されている（採用レベルは不明）
Threat Model recognized as essential (adoption level unknown)

公開脅威モデル（白書）
Public Threat Model (position paper)

コミュニケーション、構成（コンポーネントの脅威 → システム脅威）
Communication, Composition (component threats → system threats)

標準化（理解を容易にする）および機械可読性（自動化を可能にする）
Standardization (eases comprehension) & Machine readable (enable automation)

機械分析（アルゴリズム、LLM）
Machine analysis (algorithmic, LLM)

継続的な監視（コード品質、監査ログ）
Continuous monitoring (code quality, audit logs)

脅威もデルを公開しよう! Publish Your Threat Models!

メリットは危険をはるかに上回る The benefits far outweigh the dangers

Threat modeling has long guided software development work, and we consider how Public Threat Models (PTM) can convey useful security information to others. We list some early adopter precedents, explain the many benefits, address potential objections, and cite regulatory drivers. Internal threat models may not be directly suitable for disclosure so we provide guidance for redaction and review, as well as when to update models (published or not). In a concluding call to action, we encourage the technology community to openly share their PTMs so the security properties of each component are known up and down the supply chain. Technology providers proud of their security efforts can show their work for competitive advantage, and customers can ask for and evaluate PTMs rather than be told "it's secure" but little more. Many great products already have fine threat models, and turning those into PTMs is a relatively minor task, so we argue this should (and easily could) become the new norm.

PREVIEW

控えめな提案 Call to Action

ソフトウェアの設計、構築、運用に影響を与える文化的要因への意識を高める。
Raise awareness of cultural influences: software design, building, and use.

セキュリティを再考し、幅広い参加によって専門家への依存を減らす。
Rethink security; reduce expert reliance by broad participation.

脅威モデルは変化を起こすための素晴らしい方法です。今日でも始めましょう。
Threat modeling is a great way to start making a difference: get started today.

議論のための質問（又はより良い質問をして下さい）
Questions for discussion (or please ask better ones)

1. 脅威モデリングはソフトウェアセキュリティに不可欠だと確信して***いない***？
   UNconvinced threat modeling is *essential* to software security? (discussion)
2. 脅威モデリングをどのように推進するか：重要性、理論、実践、実践方法？
   How to promote threat modeling: importance, theory, practice, doing it?
3. 脅威モデリングの普及を阻む主な障害は何ですか？
   What are the major obstacles to widespread threat modeling?
4. 脅威モデリングを日本特有のソフトウェア業界にどのように適応させるか？
   How to adapt threat modeling to Japan's unique software practices?
5. 脅威モデリングをどうやって貢献できますか? How can I help?

ありがとうございました!

Thank you!

Loren Kohnfelder

https://designingsecuresoftware.com/jp