セキュリティ・バイ・デザインとアシュアランスケース

SEC研究員　金子 朋子

モノのインターネット（IoT: Internet of Things）と言われるIoTシステムは今後急激な普及・拡大が見込まれる。しかし、つながる世界は様々なリスクも抱えており、開発プロセスの早い段階から将来のハザードや脅威に備えていくことが必要とされている。またIoTシステムではつながる対象の広がりに応じて、要件はより複雑化するため、その可視化はとくに重要な課題である。そこで本稿では安全なIoT システムの枠組み作りのために開発プロセスの早い段階からセキュリティに対処する「セキュリティ・バイ・デザイン」の考え方を説明した上で、セーフティとセキュリティの要件すり合わせの意義とその手段の１つとしての「アシュアランスケース」について解説する。

1　セキュリティ・バイ・デザインとは？

1.1　現在のIoT開発の課題

現代のシステムはネットワークを介して様々な機器やクラウドと連携しながら動作している。このように異なる分野の製品や産業機械などがつながって新しいサービスを創造するモノのインターネットは新産業革命とまで言われ、大きな期待を集めている。IoTは家電、自動車、各種インフラ業者など新規プレーヤーの登場を産み、その取り込みは加速化している。しかし相互につながる際に最も懸念されるのは、IoTシステムへのセキュリティ上の脅威である。IoTシステムにおいても攻撃者はシステムの脆弱性を突いて攻撃を仕掛けてくるためである。その課題解決策として「セキュリティ・バイ・デザイン」という考え方が近年提唱されている。

1.2　セキュリティ・バイ・デザインの定義

内閣府サイバーセキュリティセンター（NISC）によるとセキュリティ・バイ・デザインの定義は、「情報セキュリティを企画・設計段階から確保するための方策」である[1]。（なお、本定義では「情報セキュリティ」としているが、「セキュリティ」とすればIoT製品やサービスにも適用できると考えられる。）

平成28年８月26日に発出された「安全なIoT システムのためのセキュリティに関する一般的枠組」においては、「将来、個々のシステムが相互に接続されることを見据え、システム相互間の接続が新たな脆弱性となる懸念があることを踏まえ、セキュリティ・バイ・デザイン（Security by Design）の思想で設計、構築、運用されることが不可欠」であるとセキュリティ・バイ・デザインの重要性が強調されている[2]。さらに「IoT システムの設計・構築・運用に際しては、セキュリティを事前に考慮するセキュリティ・バイ・デザインを基本原則とし、これが確保されていることが当該システムの稼働前に確認・検証できる仕組みが求められる。」と記述されており、安全なIoT システムのために、セキュリティ・バイ・デザインは基本原則として掲げられている。

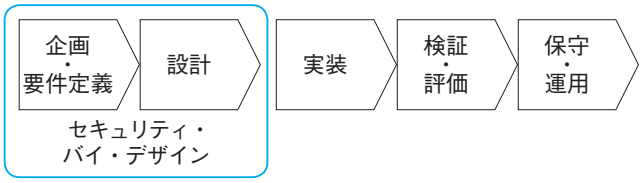


図１　セキュリティ・バイ・デザインの定義

1.3　セキュリティ・バイ・デザインのメリット

企画・設計段階という開発の早い段階からセキュリティを考慮することのメリットには、「手戻りがないため、納期を守れることや、コストも少なくできること」が考えられる。市場で運用されている段階で脆弱性が発見された場合には機器の交換やシステムの改修などが必要となるため、設計時のセキュリティ対策コストの100倍との試算もある（図２）[3]。また、他の機能ができあがってから後付けでセキュリティ対応するより、事前に対処したほうが「保守性の良いソフトウェアができること」もメリットとして挙げられる。

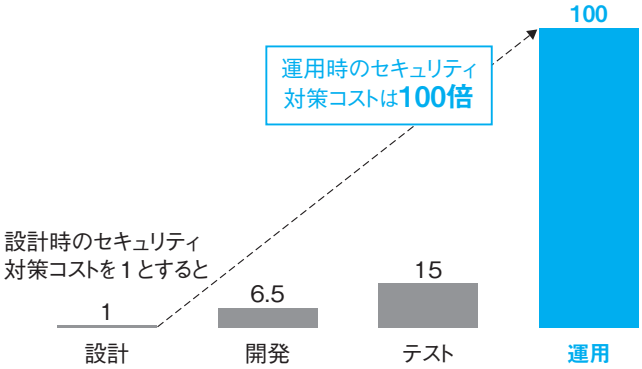


図２　開発工程別のセキュリティ対策コスト

1.4 セキュリティ・バイ・デザインの普及していない理由

それでは、なぜ、セキュリティ・バイ・デザインが普及していないのであろうか？セキュリティ・バイ・デザインが難しい理由としては以下の点が考えられる。

① セーフティ設計（設計段階で安全を作り込むこと）に比べ、セキュリティ設計（設計の段階で脆弱性の低減や脅威への対策を考慮にいれること）の歴史が浅く、上流工程の開発プロセスが定まっていない。

② 非機能要件なので、コンセプトを決める企画段階で考慮がされづらい。一般的な機能に対する要求はステークホルダ（利害関係者）の意図をシステムにより実現するのが目的であるが、セキュリティはステークホルダが実現を目的としてはいないが、当然、対応されていると考える非機能要件である。

実際、セキュリティ設計の基本方針を明文化している組織は多くはないのが現状である。「セーフティ設計・セキュリティ設計に関する実態調査結果」では、半数以上が明文化されたルールはないとしている[4]。同様の調査でセーフティに関しては自動車分野などでは明文化が進んでいることが明らかになっている。

1.5 セキュリティ開発プロセス

では上流工程の開発プロセスではセキュリティ設計として、具体的には何をすべきなのであろうか？一般的な脅威分析のアプローチは想定される脅威及び脆弱性を洗い出し、攻撃される可能性、攻撃された場合の想定被害からリスクを評価し、リスクの高い箇所にこれを抑止するための対策を検討する。具体的には①分析範囲の決定、②関係者の決定、③保護すべき資産の抽出、④前提条件の検討、⑤脅威の洗い出し、⑥対策方針の検討といった手順で検討される。

脅威分析の特徴は、セキュリティ特有の課題として、悪意の存在である攻撃者を仮定し、常に攻撃者がシステム関係者の意図しない動作をさせることを前提にリスク分析を行うことである。脅威分析は要求に基づく機能要件の分析に加えて攻撃者の存在を考慮した非機能要件の分析を必要とする。

攻撃とは、脅威を意図的に実現する手段であり、攻撃に対処できることの説明責任を果たすには、脅威分析が必要である。この脅威分析は従来の対応ではあまり実施されてこなかった施策であるが、今後のIoT時代の開発プロセスとして期待されている。

2 セーフティをまもれるセキュリティ

2.1 セーフティとセキュリティの違いと類似点

一般にセーフティとは偶発的なミス、故障などの悪意のない危険に対する安全を示すのに対し、セキュリティとは、悪意をもって行われる脅威に対しての安全を示し、セーフティとセキュリティは表1に示すように多くの違いをもっている。

表1　セーフティとセキュリティの相違点

相違点	セーフティ	セキュリティ
保護対象の違い	人命、財産（家屋等）など	情報の機密性、完全性、可用性など
原因の違い	合理的に予見可能な誤使用、機器の機能不全	意図した攻撃
被害検知の違い	事故として表れるため、検知しやすい	盗聴や侵入など、検知しにくい被害も多い
発生頻度	発生確率として扱うことができる	人の意図した攻撃のため確率的には扱えない
対策タイミング	設計時のリスク分析・対策で対応	時間経過により新たな攻撃手法が開発されるので、継続的な分析・対策が必要

両者は設計時に要件が相反することもあるので注意が必要となる。とくに生命、健康にかかわるセーフティの要件は重要である。

しかし、セーフティとセキュリティのリスク対応プロセスは類似している。セーフティではリスクの原因としてハザードを特定し、セキュリティでは脅威を特定するが、表現は異なるものの、リスクの特定、リスク分析、リスク評価、リスク対応というプロセスを繰り返すという基本的な流れは同様である。それ故、セキュリティ設計の脅威分析時に、セーフティ設計のハザードの特定と分析を行うことが可能であると考えられる。

2.2 セーフティ設計とセキュリティ設計すり合わせのメリット

IoT対応時にセーフティ設計とすり合わせをしながらセキュリティ設計を実施する手順について、考えてみよう。

インターネット冷蔵庫を新たに作成する例で考えてみると、まず冷蔵庫は既存の機能にインターネット機能を追加設計する必要が生じる。次にセキュリティを早期段階で考慮しない従来の対応ではセーフティ設計のみを実施する。そしてセーフティ設計を実施後のソフトウェア開発時にセキュアプログラミングをし、検証・評価時に脆弱性検査などを実施することになる（図3）。これに対して、セキュリティ・バイ・デザインの対応をする場合、セーフティ設計時にセキュリティ設計をすり合わせる。すり合わせをするとセーフティとセキュリティの部門間での作業の手戻りがないため、メリットが大きい。更に、IoT化することで変化する情報や通信方法の見直しだけでなく、セキュリティ上の脅威への対策を考えたインターネット冷蔵庫の企画、要件定義を実施する。事前にセキュリティ設計手順を踏むことで、セーフティとセキュリティ両方の観点からの安全性、コストなどのバランスのとれた設計を実施することが可能となる。

つまりセキュリティ・バイ・デザインはセキュリティだけのものではなく、事前にセキュリティを考慮することで、セキュリティ上の脅威にさらされるIoT機器などのセーフティも守ることができるのである。

セキュリティ・バイ・デザインはセキュリティのためだけの考え方に聞こえるが、実は「セキュリティで脅かされるセーフティも守ることができる」と言える。

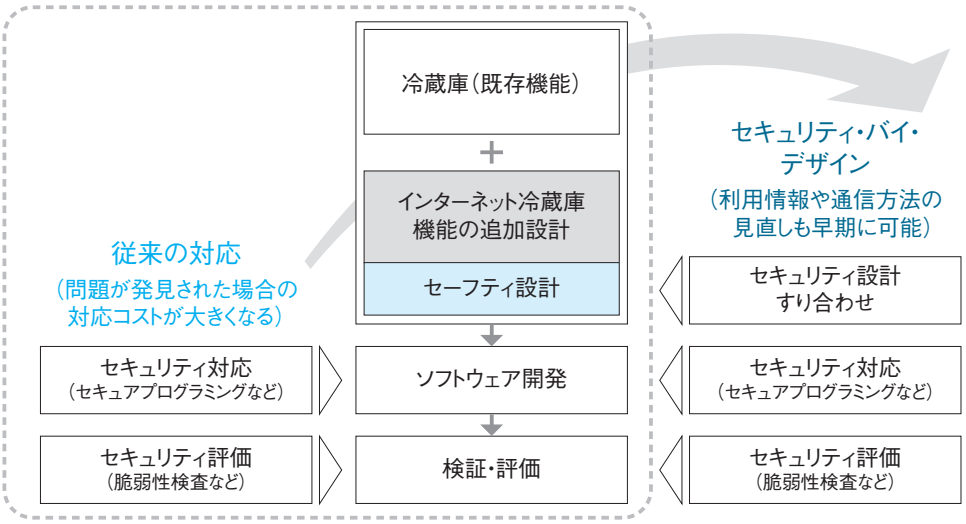


図３　IoT機器などのセキュリティ・バイ・デザインのイメージ（インターネット冷蔵庫の例）

3　アシュアランスケースとは？

3.1　ロジカルな設計品質の説明

セーフティとセキュリティを考慮した設計をしたとしても、「IoTでつながった製品を安全なものとして使って大丈夫か？」という利用者の不安に対して設計者は説明が求められる。

設計品質のロジカルな説明とは、「その設計によって目標が達成されることが、事実に基づき、論理的に説明されていること」である。

設計品質のロジカルな説明をするためには、アシュアランスケースの理論的背景となっているトゥールミン・ロジックが参考になる[5]。トゥールミン・ロジックは法律分野でイギリスの分析哲学者スティーブン・トゥールミン（Stephen Edelston Toulmin）が実社会の議論形態を分析して提唱した論証モデルである。

- 主張とは、論理として構築されるひとつの主張
- 基礎とは、論理の根拠となる、状態、事実など最初に呈示される説明情報
- 根拠とは、クレームの根拠としてデータが利用可能であることを正当化する情報

設計品質のロジカルな説明には、第三者でも分かりやすく、事実（証拠）に基づいて論理的に設計品質を説明できる「見える化」されたドキュメントが有用である。

3.2　アシュアランスケースの定義

アシュアランスケース（assurance case）とは、テスト結果や検証結果をエビデンスとしてそれらを根拠にシステムの安全性、信頼性を議論し、システム認証者や利用者などに保証する、あるいは確信させるためのドキュメントである[6]。アシュアランスは保証、ケースは論拠を意味している。

アシュアランスケースは欧米で普及しているセーフティケース[7]から始まっており、近年、安全性だけでなく、ディペンダビリティやセキュリティにも使われ始めている。アシュアランスケースはISO/IEC15026やOMGのARM[8]などで標準化が進められている。

アシュアランスケースの構造と内容に対する最低限の要求は、システムや製品の性質に対する主張（claim）、主張に対する系統的な議論（argumentation）、この議論を裏付ける証跡（evidence）、明示的な前提（explicit assumption）が含まれること、議論の途中で補助的な主張を用いることにより、最上位の主張に対して、証跡や前提を階層的に結び付けることができることである。

アシュアランスケースは対象となる機器やシステムについて、なぜその設計で目標が達成されるかを事実に基づき、論理的かつ第三者でも容易に理解できる表記で説明する手法である。

IoTの対象となる航空、鉄道、軍事、自動車、医療機器の分野の複数の安全性規格やガイドラインで要求され、欧州を中心に広く利用されている。

3.3　アシュアランスケースの表記法

「見える化」の手段としてGSN、CAE、D-Caseなどのアシュアランスケースの表記法がある（表２）。

表２　アシュアランスケースの表記法一覧

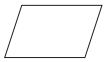
	CAE	GSN	D-Case
正式名称	Claim、Argument、Evidence	Goal Structuiring Notation	Dependability Case
登場時期	1998年	2011年	2012年
構成要素	３種類	６種類	GSNを拡張
開発組織	英Adelard社、ロンドン大学	英ヨーク大学	日本DEOSプロジェクト

代表的な表記方法は、欧州で約10年前から使用されているGSN[9]であり、要求を抽出した後の確認に用い、システムの安全性や正当性を確認することができる。日本国内ではGSNを拡張したD-CaseがJST CREST DEOSプロジェクトで開発されている[10]。要求、議論、証跡のみのシンプルなアシュアランスケースであるCAE[11]もある。

アシュアランスケースの代表的な表記法であるGSNの構成要素を表３に示す。

GSNでは前提とサブゴールに分かれる戦略の明示により論理関係を明確にした上で、各サブゴールが成り立つことで、最上位のゴールが成り立つことが保証される。

表３　GSNの構成要素

名称	図式要素	内容
主張（ゴール）		保証したいこと、命題（例：システムは安全である）ゴールは更に詳細なゴール（サブゴール）に分解される
説明（ストラテジ）		ゴールをサブゴールに分けるときの考え方（例：個別の障害ごとに議論する）
証拠（エビデンス）		ゴールが成り立つことを最終的に保証するもの（例：テスト結果、運用事例など）
前提（コンテクスト）		システムの状態、環境などゴールを議論するときの前提など（例：リスク分析の結果得られたハザードのリスト）
未定義要素		ゴールを保証するための十分な議論又はエビデンスがない（これはゴールやストラテジーにつけることができる）

3.4　アシュアランスケースによる脅威分析検討事例

本節ではIoTの具体的事例をもとにアシュアランスケースによるセキュリティ要件の可視化方法を示す。図４はスマートハウスの脅威と対策の検討例を図示したものである[12] HEMS コントローラを中心に接続されたHEMS対応機器やそれ以外のネットワーク対応機器がホームルータを介してインターネットに接続されており、外出先からスマートフォンを用いてクラウドサービス経由で家庭内の機器にアクセスすることによって、家庭内の機器の様子を監視したり、遠隔操作したりすることが可能となる。このシステムでは、スマートハウス内に設置された機器の一部に保存されたデータの漏えい、通信路上のデータの盗聴・改ざん、クラウドサービスやインターネット上に接続された中継機器への不正アクセス、（不正ログイン、その後の不正コマンド発行による許可なき遠隔操作）、クラウドサービスやインターネット上に接続された中継機器への DoS 攻撃、クラウドサービス上に保存されたデータの漏えいなどの脅威が想定される。

図４の事例をアシュアランスケースで記述したものが、図５である[13]。図５は「G_1 スマートハウスのセキュリティ設計は妥当である」というゴールを満たすために、「S_1 脅威分析の洗い出しと対策を示す」戦略を「G_2 スマートハウスの脅威の洗い出しは妥当である」と「G_3 スマートハウスの脅威に対する対策立案と選択は妥当である」の２つのゴールに分けて説明している。図５に示すG_2以下はスマートハウスにつながっている機器ごとと機器間の通信ごとに脅威を洗い出すことを求めている。各機器と機器間の通信の双方の脅威の出所をおさえれば、網羅的な脅威の洗い出しが可能になるからである。これらはG_4からG_11のゴールとして設定され、各ゴールで洗い出した脅威に対する対策をE_1からE_8の証跡として提示する。図４の事例に示された脅威の詳細が各証跡となる。

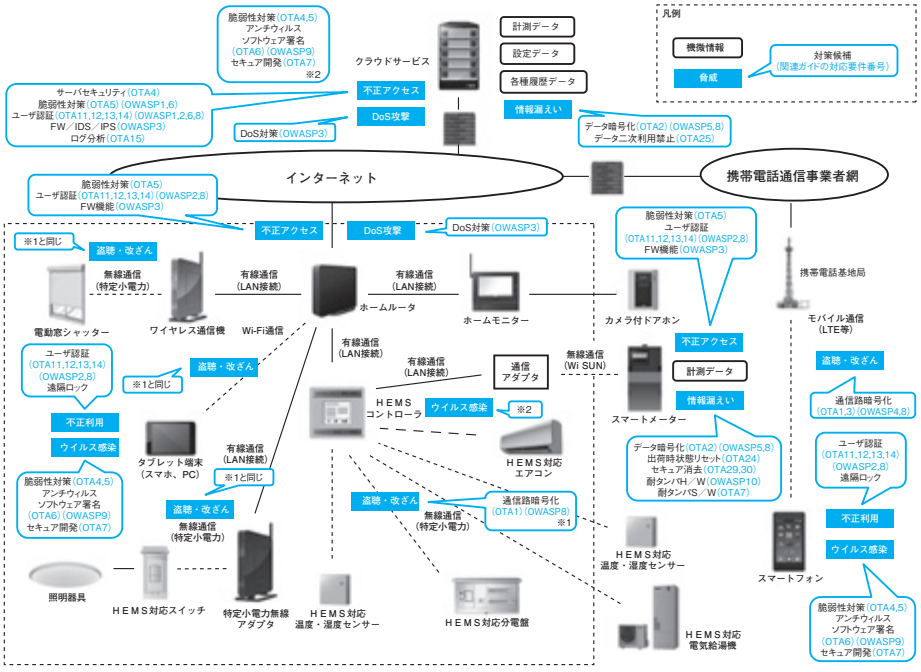


図４スマートハウスの脅威と対策の検討例

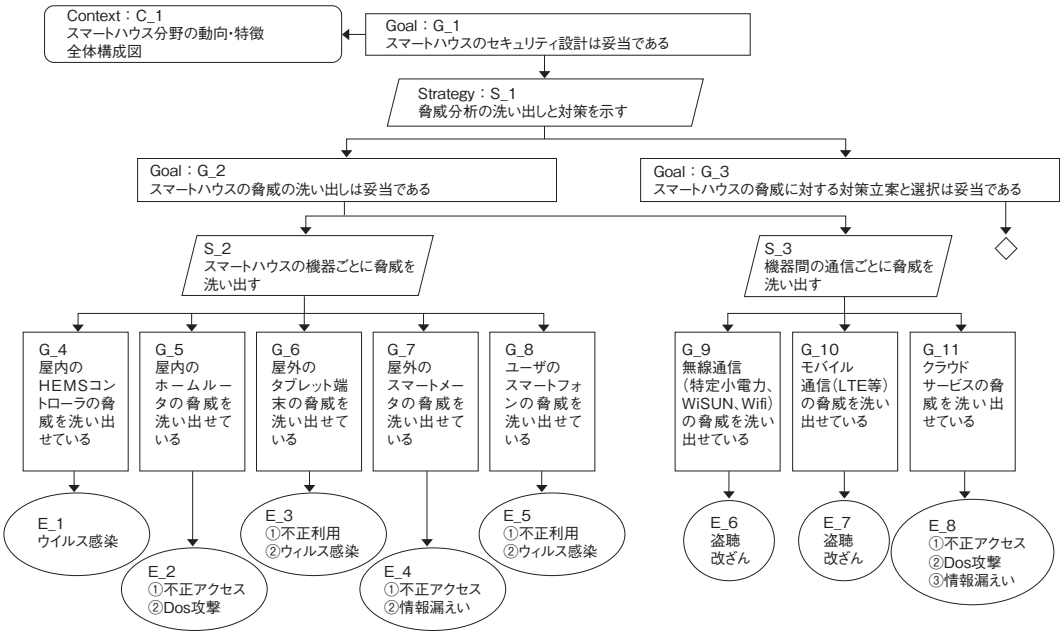


図５　スマートハウス事例へのアシュアランスケースの適用例（脅威の洗い出し部分）

図６に示すG_3以下は、洗い出した「S_4 対策ごとに分けて論証する」、「S_5 対策選択に合意する」、「S_6 残存リスクを影響分析する」という３つの戦略をプロセス化している。E_1からE_8で挙がった対策の中には、発生箇所が異なっていても対策として同じものが含まれるため、対策ごとに実施方法を証跡として示す。これらは脅威の洗い出しに対する重複の排除となる。また、実施する対策は経営層・顧客などのステークホルダとの合意が必要である。更にコストなどを考慮した実施可能な対策でなければ実施できない。そこで実施の合意を得られた対策は合意を証跡として残し、コストなどの事情で実施に至らなかった対策は影響分析をして残存リスクを証跡として示すことが必要である。これらは選択する対策と残存リスクに対処するプロセスとなる。なお、G_12からG_19のゴールが妥当である根拠としてE_9からE_16の証跡を示しているが、これらが実際に「妥当である」というためには、別の考察や判断基準が必要であろう。本提案の意義はハイレベルな脅威分析の妥当性提示である。本手法を実際に用いるためにはそのケースに応じた段階的詳細化が必要となる。本手法では脅威分析を実施したいケースをインプットとし脅威の洗い出しの結果、立てた対策がアウトプットになる、このアウトプットは運用による対処と設計による対処に分かれて実施される（図７）。設計者はアシュアランスケースを利用することで脅威対策の全体像を把握し設計できる。またトレーサビリティを保ちながら、修正、再利用をすることができる。

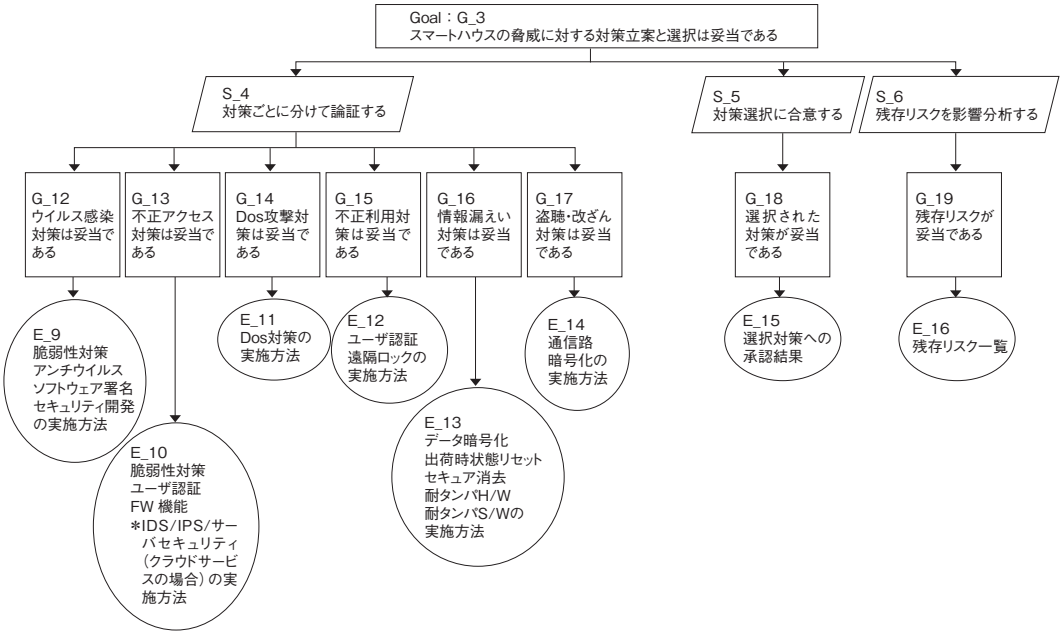


図６　スマートハウス事例へのアシュアランスケースの適用例（対策立案と選択部分）

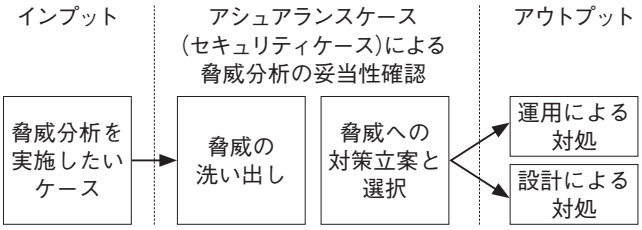


図7　脅威分析の妥当性確認

4　ステークホルダとの設計情報共有

セーフティとセキュリティの対応は企画、設計開発、販売・サポート、廃棄までライフサイクル全体において必要である。

さらに、図8のようにライフサイクルの各段階において関係する自社内の他の部門、特に品質管理部門、経営層、利用者といったステークホルダとの設計情報共有は設計品質の「見える化」のメリットの1つであり、アシュアランスケースは有効な手段となる。例えば次のような活用例が考えられる。

- セーフティとセキュリティの両部門において、設計内容を共有するためにそれぞれの部門で作成したアシュアランスケースを共有し「見える化」に利用できる。セーフティ設計とセキュリティ設計のすり合わせにも活用できる。
- ソフトウェア設計や再利用時の設計内容の理解において、新製品開発やバージョンアップ時のソフトウェア再利用時に、設計内容を理解するために活用できる。
- 設計者間での内容の理解だけでなく、経営層や品質管理部門等のステークホルダとの設計情報共有にも利用可能である。
- トレーサビリティ、説明責任のツールとして問題が発生したときに設計内容を確認したり、問題と設計との関係を説明するために活用できる。

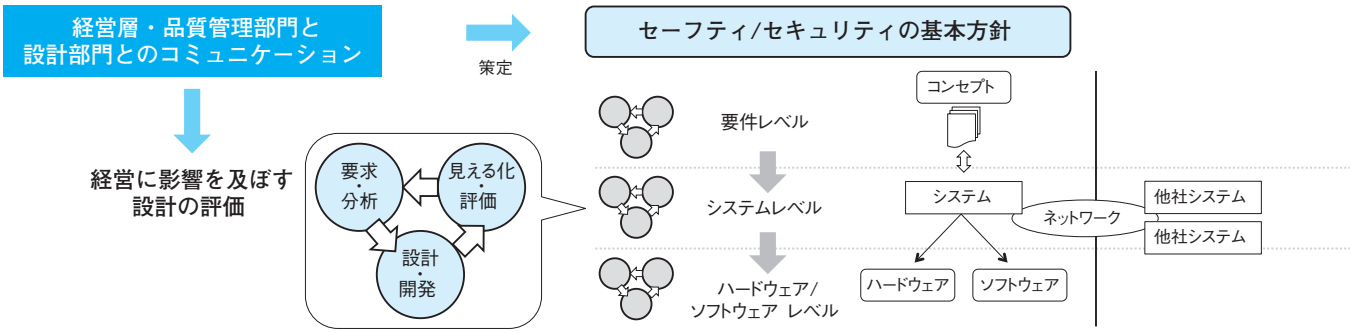


図8　ステークホルダとの設計情報共有

5　まとめ

本稿では、セキュリティ・バイ・デザインとは「情報セキュリティを企画・設計段階から確保するための方策」であり、開発の早期段階からセキュリティを考慮していくことの必要性を解説した。またつながる対象の増加によってセキュリティ上の脅威がセーフティを脅かす可能性の高まるIoTの開発において、セキュリティをセーフティと共に考慮していくことによりセーフティを守れるセキュリティとすべきであることを示した。

更に、設計品質のロジカルな説明とは、「その設計によって目標が達成されることが、事実に基づき、論理的に説明されていること」であり、その有効な手段の1つとしてアシュアランスケースが利用されていることを説明し、セキュリティ上の脅威分析にアシュアランスケースを用いた事例を示した。

IoT時代の設計は、ステークホルダとの設計情報共有がカギを握る。ここにアシュアランスケースの利用を通じて、設計品質のロジカルな「見える化」が普及していくことを念願している。

参照文献

[1] NISC, http://www.nisc.go.jp/conference/seisaku/dai15/pdf/15siryou02.pdf
[2] NISC, 安全なIoTシステムのためのセキュリティに関する一般的枠組, http://www.nisc.go.jp/active/kihon/res_iot_fw2016.html
[3] IPA, つながる世界のセーフティ＆セキュリティ設計入門
[4] IPA, セーフティ設計・セキュリティ設計に関する実態調査結果
[5] IPA, アシュアランスケース入門, 2015, http://www.ipa.go.jp/files/000043906.pdf
[6] 松野裕、高井利憲、山本修一郎「D-Case入門 ～ディペンダビリティ・ケースを書いてみよう！～」, 2012
[7] T P Kelly & J A McDermid, “Safety Case Construction and Reuse using Patterns”, in Proceedings of 16th International Conference on Computer Safety, Reliability and Security (SAFECOMP'97), Springer-Verlag, 1997
[8] OMG, ARM, http://www.omg.org/spec/ARM/1.0/Beta1/
[9] Tim Kelly and Rob Weaver, The Goal Structuring Notation – A Safety Argument Notation, Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, 2004
[10] 松野裕、山本修一郎：実践　D-Case ～ディペンダビリティケースを活用しよう！～, 株式会社アセットマネジメント、2013
[11] The Adelard Safety Case Development (ASCAD), Safety Case Structuring: Claims, Arguments and Evdence, http://www.adelard.com/services/SafetyCaseStructuring/index.html
[12] IPA, IoT開発におけるセキュリティ設計の手引き, 2016
[13] 金子朋子、髙橋雄志、勅使河原可海、田中英彦：CC-Caseを用いたIoTセキュリティ要件の可視化, 2016