複雑なネットワークソフトウェア開発運用のためのトラストを考慮したセキュリティ＆プライバシ・エコシステム

代表研究者	鷲 崎 弘 宜	早稲田大学　教授
共同研究者	吉 岡 信 和	国立情報学研究所　准教授
共同研究者	櫨 山 淳 雄	東京学芸大学　教授
共同研究者	大 久 保 隆 夫	情報セキュリティ大学院大学　教授

1 はじめに

サービスやデータの集中管理、自然災害や侵入者といった外的要因の変化や増大、さらには設計思想やセキュリティの異なる多様な機器の接続に伴い、攻撃やデータ漏洩のリスクは増大している。また、他の品質を維持したまま必要なセキュリティおよびプライバシを確保することは、社会的急務である。[IPA]では IoT 環境におけるセキュリティやプライバシ対策・考慮の重要性が指摘され、[And15]ではクラウド環境におけるセキュリティの確保の重要性が指摘されている。本研究の成果を活用することで、IoT やクラウドに代表される複雑なネットワーク・ソフトウェアシステムにおいて、多種多様、そして多量なデータをリアルタイムに扱い業務の飛躍的な効率化や新たなサービス・価値の創造に寄与できることが期待される。さらに、セキュリティやプライバシの前提条件として、提供・利用間におけるトラスト（信頼関係）の事前構築が不可欠である。

そこで本研究では、以下の研究経緯の下で、クラウド等の複雑なネットワークソフトウェアに対し過去成果や知識を再利用してトラストに基づくセキュリティとプライバシを組み入れることで次への開発運用へと役立てる共存・循環・進化型エコシステムの枠組みを実現した。

本報告では以降において、2 節で研究経緯を説明する。3 節において研究成果を説明し、4 節ではその一環として特にセキュリティパターン研究の分類体系について説明する。同様に研究成果の一環として、5 節では脆弱性記述の扱い、6 節ではモデルベース適用・アシュアランスケースの仕組みを説明する。最後に 7 節において、本研究の成果がもたらす価値や今後の展望を説明する。

2 研究経緯

研究経緯としては、代表者らは 2015-2016 年度にセキュリティに関する代表的な既存メタモデル群を統合し、かつ、クラウドサービスに依存する部分と非依存な部分を明確とした[Was16]。得られたセキュリティのメタモデルをプライバシおよびトラスをも扱う形で改訂の上、本研究の基礎として用いた。

また代表者らはセキュリティパターンや関連技術について世界的な研究実績をあげて国際的に研究領域をリードする立場にある。具体的には、セキュリティパターンの分類手法[Fer08]、モデルベースの適用・検証手法[Kob14]、パターン検索システム[Kub08]等を実現している。これらを応用して、トラストおよびプライバシも含めて事例やパターンを扱う知識ベース、および、知識間の関連を通じた検索、全体・詳細閲覧および管理を容易に可能とする知識ベース管理システムの基礎を実現した。

3 研究成果

3-1 全体像

本研究では、企画から開発、運用にいたるライフサイクル中のセキュリティ、プライバシ、トラストに関わる成果物や知識等について、メタモデル上で整理体系化・統合・再利用することで新たな高信頼ネットワーク・ソフトウェアシステムを進化的に生み出すエコシステムの枠組みを実現した。そのうえで、ケーススタディにより、エコシステムの枠組みが運用上の新リスクや攻撃・対策を、当該および他システムの企画・開発・運用へ役立てられることを確認した。研究の全体像を図 1 に示す。

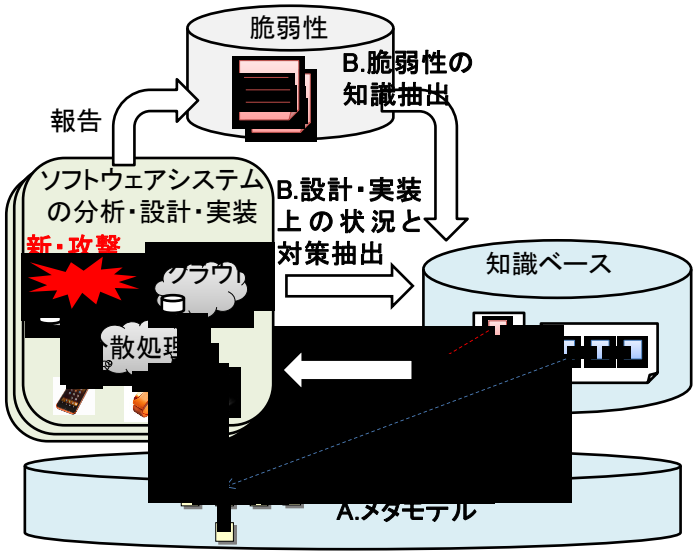

図 1. 実現した研究成果の全体像

3-2 メタモデル（A）

そこで本研究では、密接に関係し合うセキュリティ、プライバシ、トラストの概念を、IoT やクラウドに代表される複雑なネットワーク・ソフトウェアシステムの文脈上で整理したメタモデルを構築することを目標とした。

その達成のために、A1. 既存のエコシステムおよび関連する脆弱性・脅威公開状況の調査、および、A2. 調査結果を組み入れた形での申し込み者の既存メタモデルの拡張した新たなメタモデルの構築を進めた。メタモデルとしては調査結果を組み入れた形で、主要概念と概念間の関係を整理したメタモデルの構築に成功した。メタモデルの抜粋を図 2 に示す。調査過程で、メタモデル、メタモデル上に構築する知識ベースおよびそれらの扱いに必要な追跡技術に関する関連研究に多く行き当たり、そのサーベイの結果としての A3. 特徴の整理・体系化を進めた。

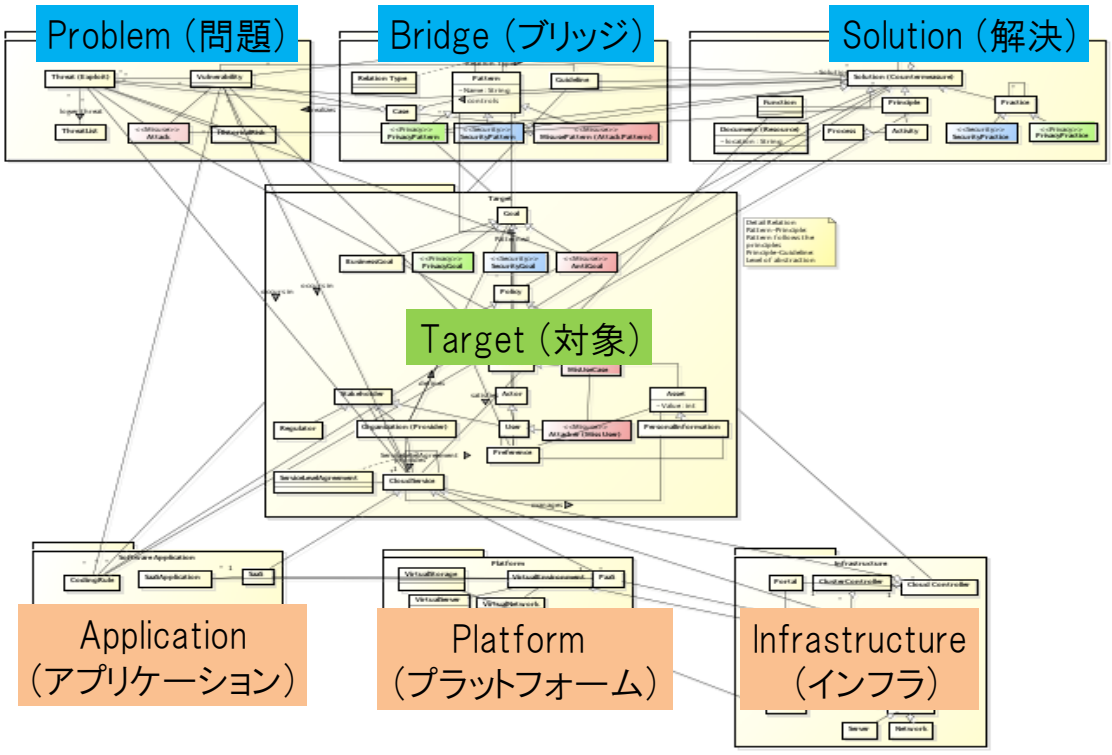

図 2. 実現したメタモデル（抜粋）

3-3 知識ベース（B）

そこで本研究では、過去の脆弱性や対策およびその発生時点の関係の報告から、共通部分をセキュリティ、プライバシ、トラストのパターン等の知識とし、メタモデル上で整理する技術を実現し、メタモデルに基づく知識ベースを構築することを目標とした。その達成のために、B1. 知識ベースの基本アーキテクチャの設計をほぼ予定通りに完了し、続いて、B2. 知識ベースの半自動構築を進めた。さらに、B3. 評価実験を通じた知識ベースの有効性の一部の検証に成功した。

3-4 モデルベース適用（C）

メタモデル上に整理された知識群を、知識ベース上でトラストの状況および要求に基づき問い合わせ検索し、設計モデルレベルで組み合わせて新システムを開発拡張することを支援することを目標とした。その達成のために、C1. 適用の仕組みを基本設計し、C2. 知識群の適用支援、および、C3. 有用性検証・改善をモデルベースに進める手法を実現した。

4 セキュリティパターン研究の体系（A3 の一部の詳細）

セキュリティパターン研究の分類の基盤として、セキュリティやセキュリティパターンおよびその活用に特有の概念を整理する必要がある。本研究では、クラウドサービス開発運用におけるセキュリティおよびプライバシ知識を整理した上述のメタモデルにおいてクラウドに依存せず、セキュリティパターン研究に関連する概念を図 3 に整理した。

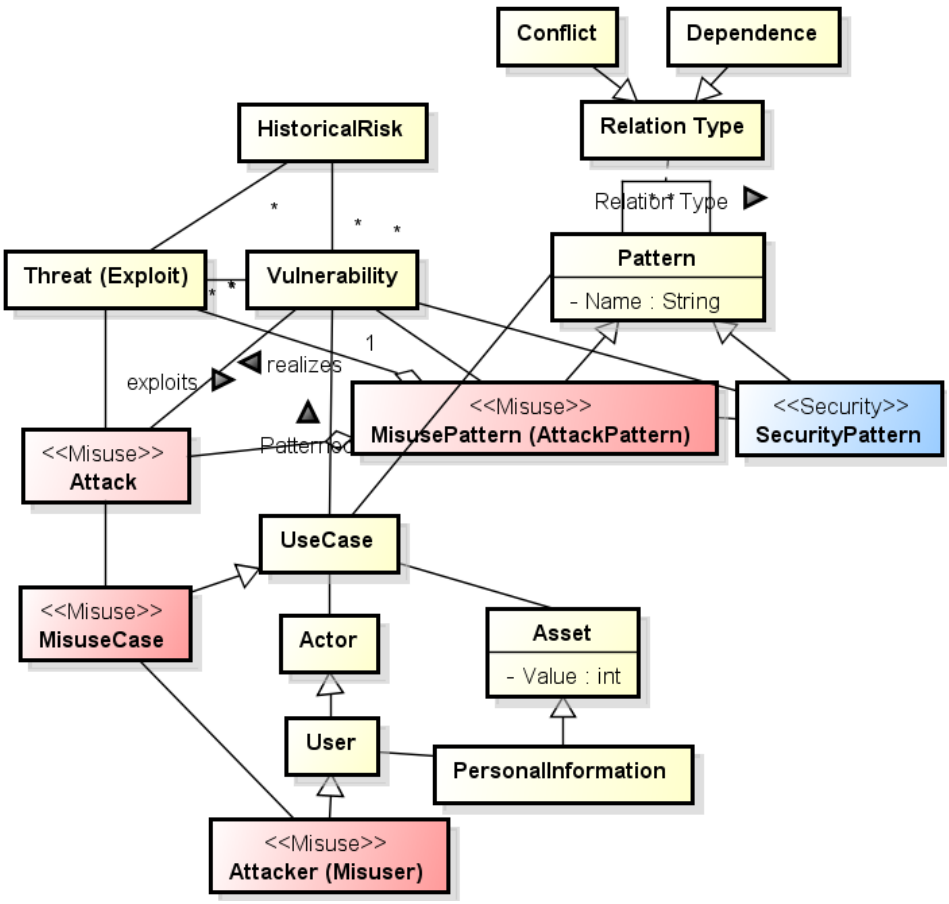

図 3: 概念モデル

そして前述の概念モデル中の主要な概念に基づき、セキュリティパターン研究の分類上のファセットとして以下を識別した。それらを体系化した結果を図 4 に示す。本研究ではセキュリティ知識について、この体系に基づいて整理された各種の知識を用いた。

(1)研究の目的: 当該セキュリティパターン研究が扱う話題（トピック）、対象とするシステム・ソフトウェアライフサイクル上の段階（フェーズ）、および、研究成果の利用者（ユーザ）が挙げられる。
(2)研究成果の実現方法: セキュリティパターン研究成果を実現するプラットフォーム、成果を自動化してツールとしてまとめているかどうか、成果の適用や導入によりセキュリティに変化があることを測定する仕組みを取り入れているかどうか、および、成果のもともとの研究目的に照らした評価をケーススタディや実験などで実施しているかどうか、が挙げられる。
(3)品質特性: セキュリティパターン研究が扱う品質特性関連の分類として、セキュリティ特有の問題として扱う脆弱性・脅威、および、情報の機密性、完全性、可用性に代表されるセキュリティ特性が挙げられる。
(4)セキュリティ関連パターン: セキュリティパターン研究が扱うセキュリティパターン、アタックパターン、ミスユースパターンの種別が挙げられる。
(5)セキュリティ手法: セキュリティパターン研究の手法上の特徴として、方法論やモデリング手法、さらにはパターン間関連の扱いの有無が挙げられる。

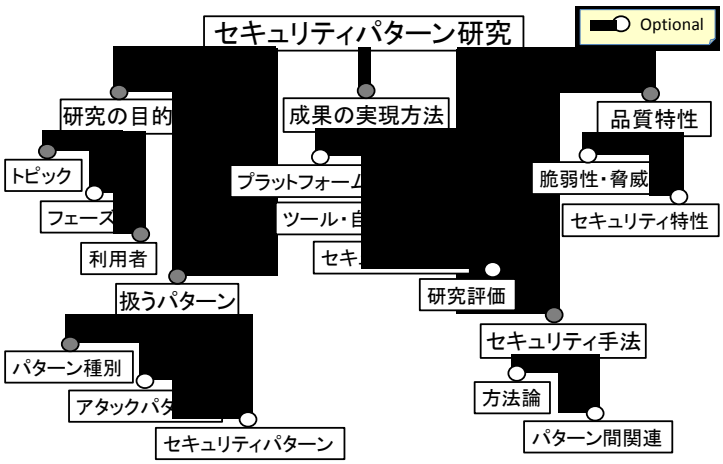

図 4: セキュリティパターン研究の分類体系

5 脆弱性記述の扱い（B2 の詳細）

日々アップデートされる Common Vulnerabilities and Exposures (CVE), National Vulnerability Database (NVD), Japan Vulnerability Notes (JVN)等の代表的な脆弱性データベースやそれらからリンクされた個々の対策報告から、共通部分をパターン等の知識として抽出し、他との関連付けによる体系化を経てグラフベースの知識ベースを構築および更新する方法を設計した。
具体的には、各データベースや報告集の記述様式はまちまちであり、メタモデル中の語彙へとマッピングして一貫して扱うことを実現した。また、各データベースや報告集の中では、例えば図 5 は CVE における異なる脆弱性の記述であるが、to に続いた脅威（破線）の指定、via に続いた攻撃手法（下線）の指定など、暗黙の記述規則を見て取れる。その明示的な規則化によりメタモデルへの展開を実現した。

CVE-2016-2000	CVE-2016-0790
HPE Asset Manager 9.40, 9.41, and 9.50 and Asset Manager CloudSystem Chargeback 9.40 allow remote attackers to execute arbitrary commands via a crafted serialized Java object, *related to the Apache Commons Collections (ACC) library*.	CloudBees Jenkins before 1.650 and LTS before 1.642.2 *do not use a constant-time algorithm to verify API tokens*, which makes it easier for remote attackers to determine API tokens via a brute-force approach.

図 5: CVE における記述規則の特定例

6 モデルベース適用とアシュアランスケース活用（C2, C3 の詳細）

メタモデル上に整理された知識群を、知識ベース上で要求に基づき問い合わせて検索し、組み合わせて新

ソフトウェアシステムを開発あるいは拡張することを支援する手法を実現した。ここで、代表者らの過去のセキュリティパターン適用・検証技術[Shi10][Kob14]を発展応用し、ソフトウェアシステムの分析・設計モデルに対するモデル変換を通じた知識群を組み合わせた適用を基本とした（図 6)。

適用にあたりセキュリティ＆プライバシ要求に対する対策の統合的適用、その実現結果としての設計、実装が有効な根拠の表明として、知識ベース中の知識を参照するアシュアランスケースの記載も行った。これにより、以降の運用でリスクや攻撃が発生した際に、アシュアランスケース上で誤りや新たな脆弱性・問題を識別し、追跡可能な分析、設計、実装上で対策を検討すると同時に、その脆弱性・対策を報告しやくする。さらには知識ベース中の知識の更新を通じて高信頼化・最新化を支援する。

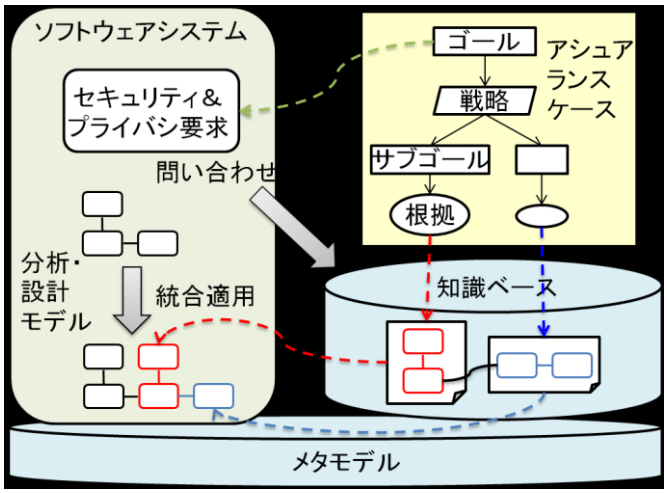

図 6. モデルベース適用とアシュアランスケース活用

メタモデルを用いて一つのパターンの User や Attacker、Provider によるユースケースとそれに関連する目的等と Cloud Service とのつながり方を以下に XSS(クロスサイトスクリプティング)を題材にしたメタモデルの活用例を用いて示す。図 7 はメタモデルを用いた XSS による攻撃者(Misuser)が顧客の個人情報を得るといった目的(Anti Goal)を達成するクラス図である。

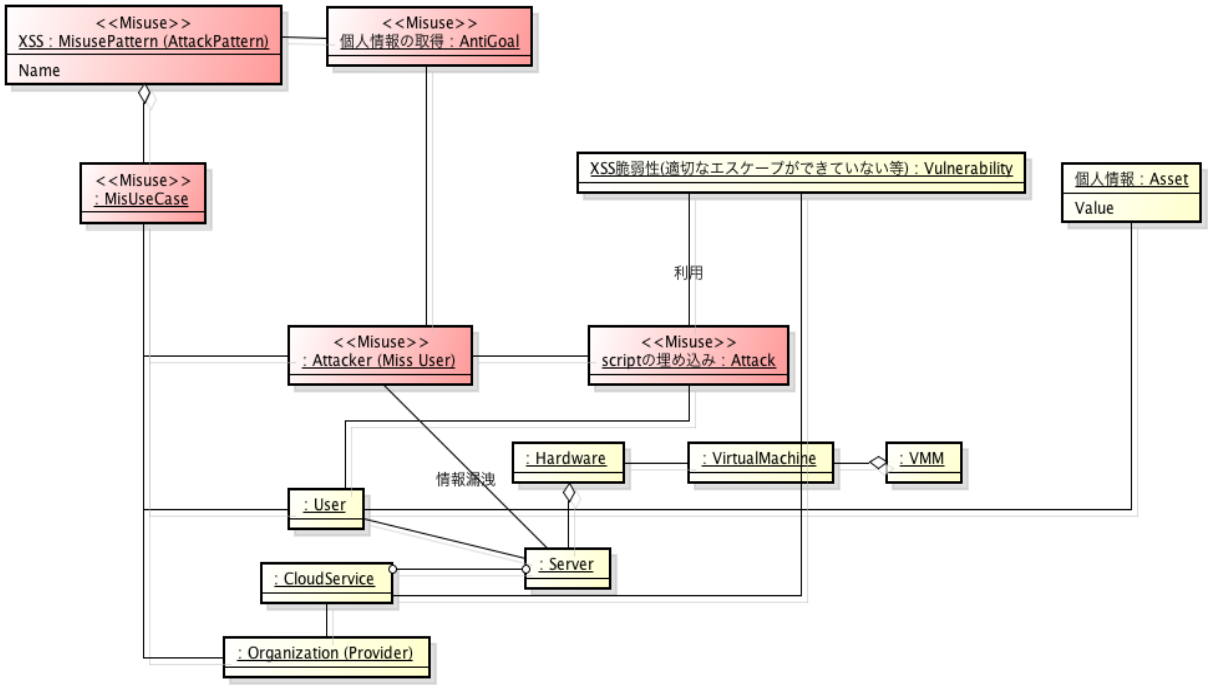

図 7：メタモデルを用いた XSS による攻撃例

この図は Attacker(攻撃者)が XSS という Misuse Pattern を用いて、XSS の脆弱性をもったクラウドサービスへ攻撃を行い、Cloud Service 内にあるユーザの個人情報を手に入れる流れを示している。ここに示されている Cloud Service がコアパッケージである User や Provider 等と Infrastructure 階層のパッケージ内にある Server や Hardware 等とを結びつけている。本メタモデルにより XSS のパターンによって影響を受ける関係性が明確となる Misuse Case を表すことができ、対策を考えるときの全体像として用いることができる。これはクラウドサービス攻撃者からの視点となる Misuse Case だが、同様に通常の Use Case であるセキュリティ側のクラス図も本メタモデルを用いて表現することができる。この XSS 脆弱性を Security Pattern の Sanitizing(無効化)を用いて対処する Use Case として以下の図 8 のメタモデルを用いた XSS 脆弱性の対処例を示す。

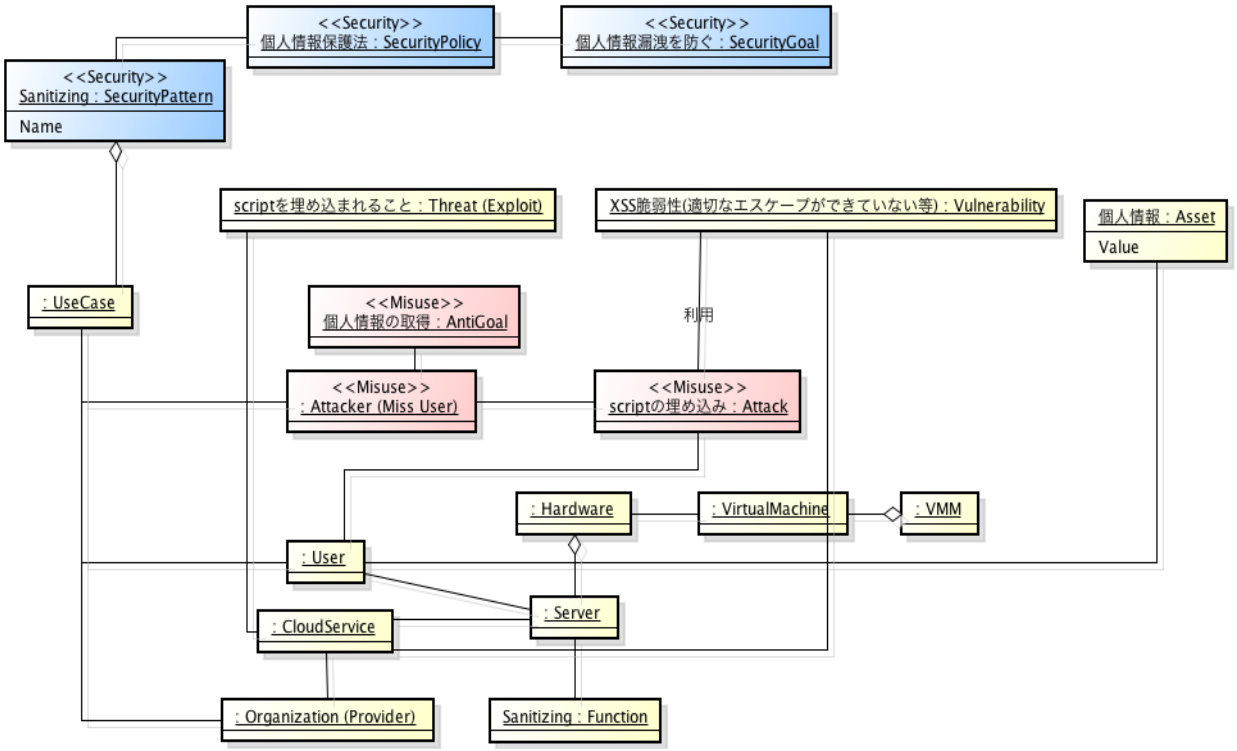

図 8：本メタモデル用いた Sanitizing による XSS 対処例

この図では Provider が個人情報保護法に基づきユーザの個人情報漏洩を防ぐために XSS 脆弱性のある Cloud Service に Security Pattern の Sanitizing(無効化)を行いサーバ内で不正なスクリプトをはじく処理を行い、User の個人情報の漏洩を防ぐことを実現している。これにより図 8 で示した XSS の攻撃を防ぐことができ、Security Pattern を適用したときに関連するクラウドサービスへの影響などの全体像も明確にすることができる。このように本メタモデルは User や Attacker と Provider、Cloud Service が様々な目的やポリシーをもち、脅威や脆弱性などと関連してパターンやプラクティス、Use Case、Misuse Case を表現、分析することができる。また、それらを表現することが新たなパターンを生み出す基盤ともなりえ、パターンやプラクティスの関連要素の整理を行うことにより今後のパターンの発展にも貢献することができる。

メタモデルを基に XSS(クロスサイトスクリプティング)の全体の動きをシーケンス図によって表現し、その表現の方法も 3 パターンに分けて表現する。まずは通常の Misuse パターン XSS のみを扱うシーケンス図を図 9、そしてそれを Security パターンの Sanitizing によって解決する図を図 10 に示す。

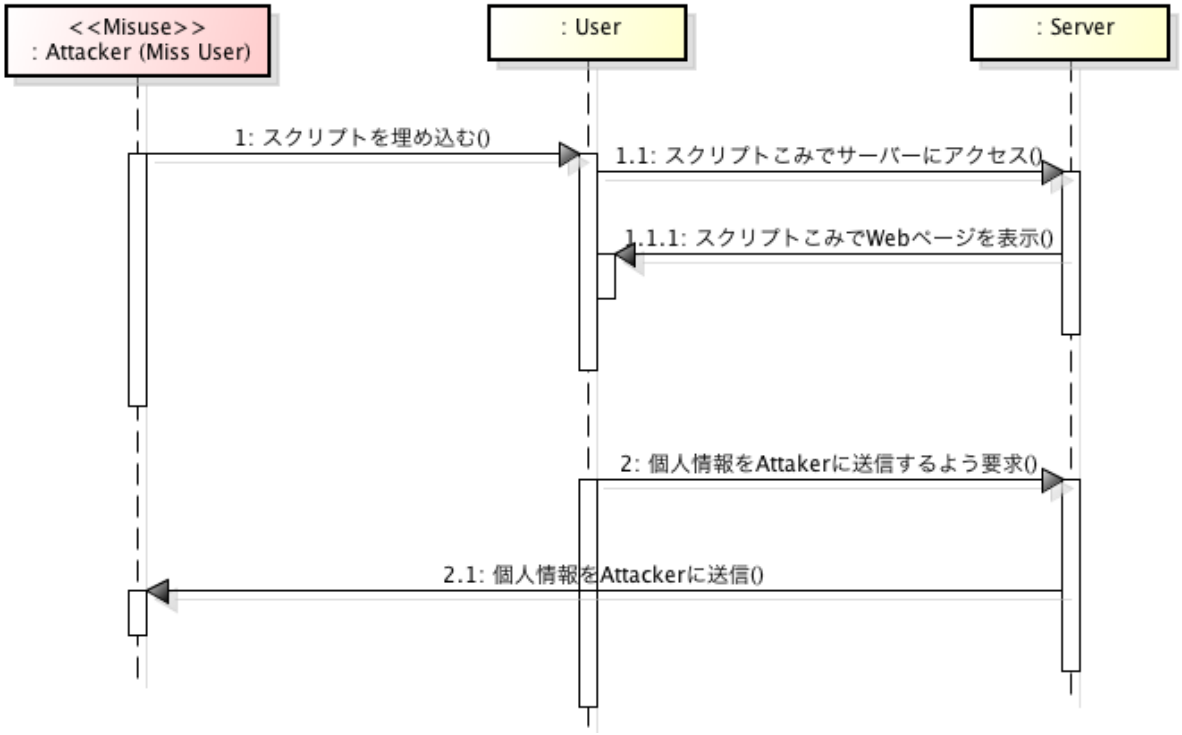

図 9：XSS のシーケンス図

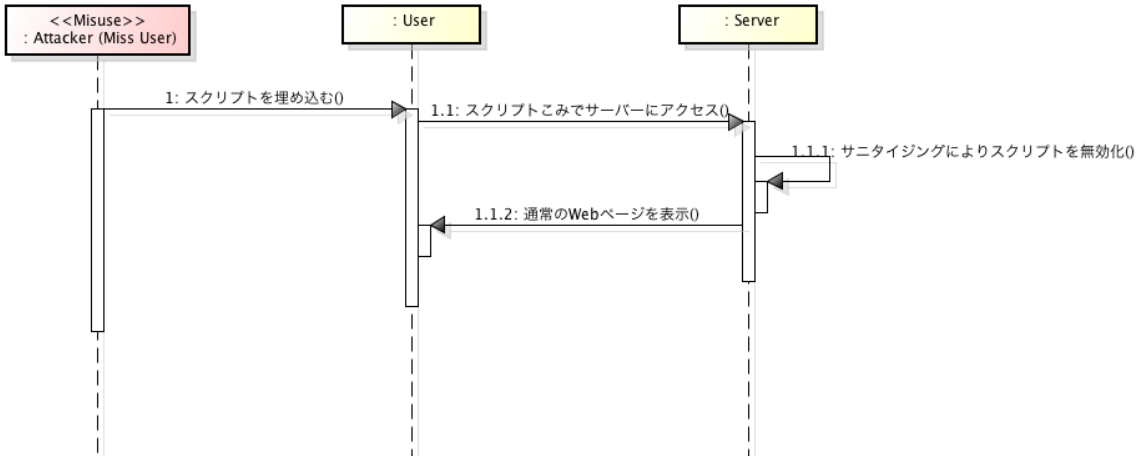

図 10：Sanitizing による XSS 対処のシーケンス図

図 9 に示すように XSS とはまず、Attacker が User に対してスクリプトを埋め込んだリンクを送り、User がそのリンクを踏んでしまうことで、User 自身がスクリプトを埋め込んだアクセスを行ってしまい、スクリプトの内容によってはたとえアクセスした先は正規のアドレスでも Web ページ自体を本来のものとは大きく改ざんすることができる。そして Attacker はスクリプトによりフォームを作成しその宛先を Attacker 自身のサーバに指定すれば User は気づかずそこで個人情報を入力してしまい、その入力情報は Attacker に送られることとなってしまうものである。この XSS の問題はサーバ側に XSS の脆弱性をもっていること、つまりスクリプトを埋め込まれた際にそのままスクリプトをサーバで実行してしまうことが原因である。そしてその対処方法としてここでは Sanitizing による対処方法を図 10 に示す。図 10 のように User がスクリプト込みでサーバにアクセスしてきた際、Sanitizing によってスクリプトを意味する<、>等の文字を別の文字へと変換するコードを書いておけば、スクリプトが実行されることを無効化することができる。そうすることで User には通常の Web ページが表示され、そこで個人情報を入力したとしても通常通りの扱いを行うことができる。このパターンをひとつの要素としてより高い抽象度で XSS の Misuse パターンによる影響を示した図を

図 11 に示す。また Sanitizing による Security パターンによる対処を行ったときの図を図 12 に示す。

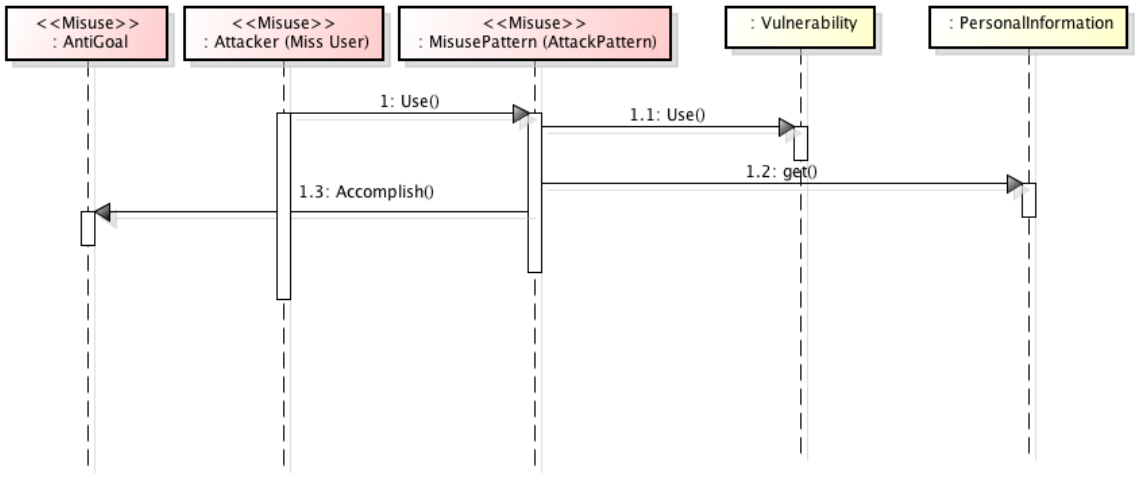

図 11：XSS を一つの項として扱ったときのシーケンス図

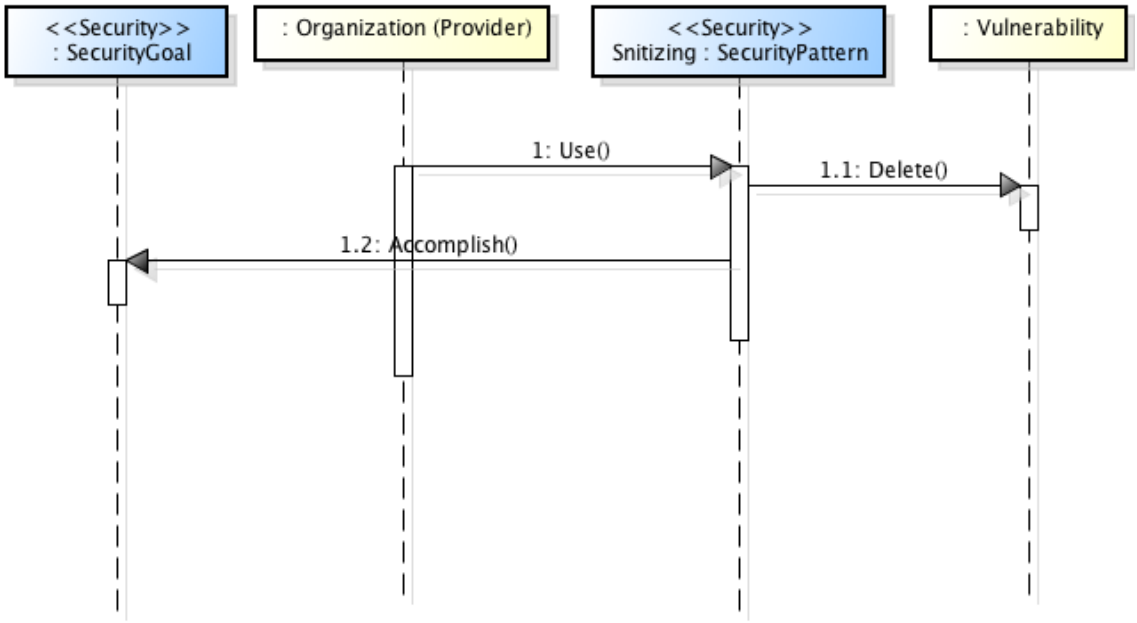

図 12：Sanitizing を一つの項として扱ったときのシーケンス図

図 11 によって Attacker が XSS という Misuse パターンを用い、脆弱性を利用して個人情報を手に入れるといった一連の流れが表現されている。そして図 14 により Provider(クラウドサービス開発・運用者)が Sanitizing による Security パターンを用いて XSS の脆弱性をなくすことで Security Goal が達成されている。そしてこれらの XSS や Sanitizing といったパターンの詳細を示したものが先ほど示した図 9、図 10 である。最後に XSS や Sanitizing を一つの項とはせずにパターンを表現し、そのパターンの全体との関連も表す先ほどの図 9 と 11、図 10 と 12 をそれぞれ合わせたような抽象度のことなる概念をまとめた図を XSS の Misuse パターンシーケンス図として図 13、Sanitizing の Security パターンとして図 14 に示す。

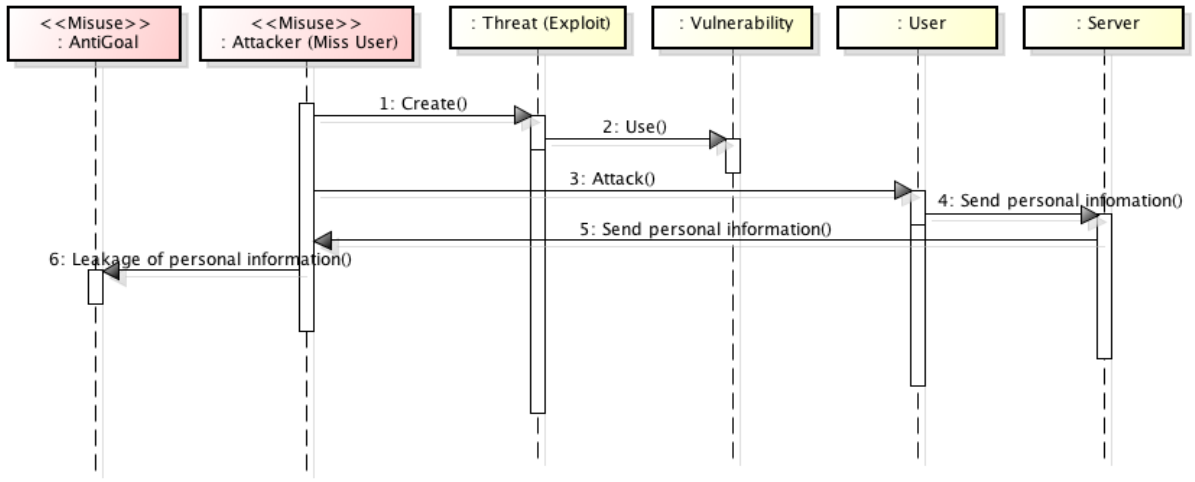

図 13：異なる抽象度をもった XSS のシーケンス図

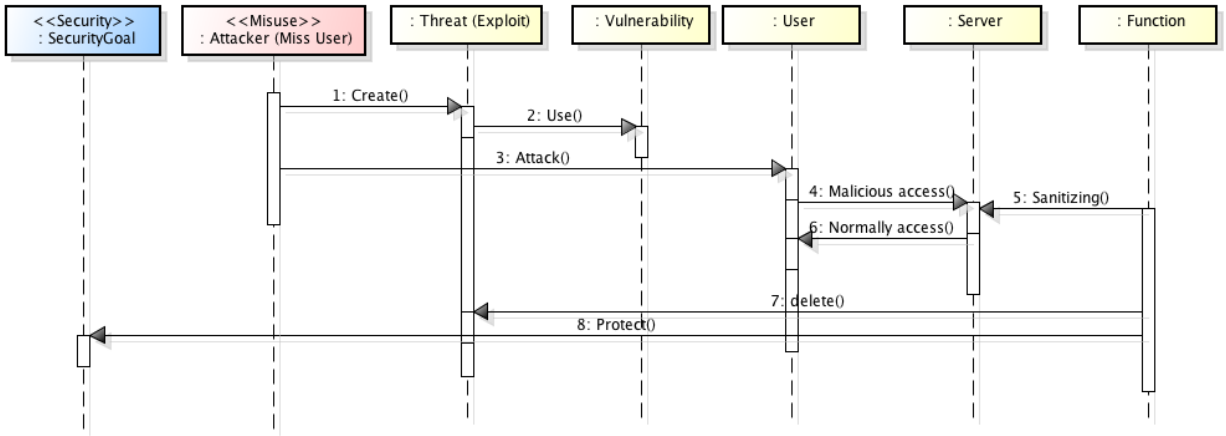

図 14：異なる抽象度をもった Sanitizing のシーケンス図

図 13 では XSS の一連の流れを表しながら Anti Goal や Vulnerability、Threat への関連を示している。これにより抽象度が異なるが一つの図でパターンの詳細から全体の流れまでを表すことができ全体像を漠然と捉えることができる。図 13 でとりあげている XSS の Sanitizing による対処方法を図 14 で示している。図 14 についても Sanitizing による流れから Security Goal の達成、Threat の削除まで表現されていて全体の大まかな流れを把握するのに適している。これらのようにメタモデルを用いて様々なパターンやプラクティスをクラス図やシーケンス図をもちいて異なる抽象度での表現することができるため、クラウドサービス開発・運用者の要求に沿った抽象度でシステムやパターンを理解することができる。これはクラウドシステムを開発するときはもちろん、運用中や保守においても有用である。

7 おわりに

本研究では、以下の研究経緯の下で、クラウド等の複雑なネットワークソフトウェアに対し過去成果や知識を再利用してトラストに基づくセキュリティとプライバシを組み入れることで次への開発運用へと役立てる共存・循環・進化型エコシステムの枠組みを実現した。

本研究の成果を用いることで、複雑なネットワーク・ソフトウェアシステムの開発者・運用者は、知識ベース中の過去の脆弱性やパターンを参照して効率的・効果的に、一定のトラスト構築の上でセキュリティやプライバシの要求を獲得・分析し、知識ベース中で要求に対応したリスク対策候補を識別し設計・実装できる。知識ベースと共通のメタモデルに基づき設計・実装することで、対策群を適切な箇所へ適用および組み合わせることを支援、さらに要求と実現箇所の対応を容易に追跡でき、一定のトラスト上のセキュリティとプライバシにおける既知のリスク対策の確保を一貫かつ効率的に進められると同時に新リスクへの対策も容

易となる。

【参考文献】

[IPA] IPA セキュリティセンター, IoT 開発におけるセキュリティ設計の手引き, 2016

[And15] S.J. Andriole, Who Owns IT?, CACM, Vol. 58, No. 3, pp.50-57, 2015

[Was16] H. Washizaki, et al, A Metamodel for Security and Privacy Knowledge in Cloud Services, IEEE SERVICES'16

[Fer16] E. B. Fernandez, N. Yoshioka, H. Washizaki, Modeling and Security in Cloud Ecosystems, Future Internet, Vol. 8, No. 13, 2016

[Fer08] E.B. Fernandez, H. Washizaki, et al., Classifying security patterns, APWeb 2008

[Kob14] T. Kobashi, N. Yoshioka, H. Kaiya, H. Washizaki, T. Okubo, Y. Fukazawa, Validating Security Design Pattern Applications by Testing Design Models, IJSSE, Vol. 5, Issue 4, pp.1-30, 2014

[Kub08] A. Kubo, H. Nakayama, H. Washizaki, et al., PatternRank: A Software-Pattern Search System, PloP

[Shi10] Y. Shiroma, et al., "Model-Driven Security Patterns Application and Validation," PLoP 2010

〈発 表 資 料〉

題　　名	掲載誌・学会名等	発表年月
Cloud Security and Privacy Metamodel: Metamodel for Security and Privacy Knowledge in Cloud Services	6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018)	2018 年 2 月
セキュリティパターン研究の分類体系と文献調査	情報処理学会第 198 回ソフトウェア工学研究発表会	2018 年 3 月
An abstract security pattern for Authentication and a derived concrete pattern, the Credential-based Authentication	7th Asian Conference on Pattern Languages of Programs (AsianPLoP 2018)	2018 年 3 月
Security Patterns: Research Direction, Metamodel, Application and Verification	The 2017 International Workshop on Big Data & Information Security (IWBIS)	2017 年 9 月
Pitfalls and Countermeasures in Software Quality Measurements and Evaluations	Advances in Computers	2017 年 6 月