博士論文

産業制御システムのサイバーセキュリティ対策を
自動化するオントロジ駆動型モデリングに関する研究

A Study on Ontology-Driven Model Engineering based on
Automated Cybersecurity Processes for Industrial Control Systems

藤田 淳也

2024 年 9 月

電気通信大学大学院 情報理工学研究科
機械知能システム学専攻

博士論文審査委員会

主任指導教員　主査　　金子 修　　教授

委員　澤田 賢治　　准教授

委員　小木曽 公尚　　教授

委員　清 雄一　　教授

委員　新 誠一　　名誉教授

Abstract

In factory automation and process automation systems, control systems responsible for monitoring and controlling machinery have evolved in tandem with advancements in computer technology. This evolution has included the integration of IoT capabilities into control system components and the convergence of information technology and operational technology, which has consequently introduced numerous security vulnerabilities. Minimizing these security risks has become a significant societal challenge. Implementing security measures in control systems not only requires specialized knowledge inherent to these systems but also demands an understanding of the latest trends in cyberattacks and defensive technologies. Particularly during the planning of security measures, the high cost of implementing threat modeling practices, which are essential for devising effective control system security specifications, has been a barrier to advancing security measures in control systems.

The purpose of this dissertation is to develop a system that helps control system managers streamline threat modeling by improving the reusability and automation of threat modeling knowledge, eliminating the impersonal nature of the process, and reducing the cost of design. This paper proposes "ontology-driven model engineering" based on a mathematical representation of elements related to threat modeling. Threat modeling encompasses the processes of risk identification, risk assessment, and risk mitigation. Methodologies and algorithms based on ontology-driven model engineering for challenges unaddressed by existing methods in each of these processes have been suggested.

During the risk identification phase, we proposed a modeling technique for attackers to prioritize threat scenarios, building on previous research focused on the comprehensiveness of threat scenarios. In the risk assessment phase, compared to traditional methods that solely focus on vulnerabilities, we introduced a risk assessment model that quantitatively expresses risks based on the internal structure of threat scenarios centered on attackers. Finally, in the risk mitigation phase, we proposed a method that allows for the prioritization of security measures, even without known cost information pertaining to security strategies automatically.

Applying the proposed methods to each phase of threat modeling demonstrated the potential to resolve issues that cannot be addressed by existing methods. This dissertation presents a framework that can realize the reuse of knowledge and automation in threat modeling through ontology-driven model engineering, suggesting the feasibility of achieving the objectives of this research.

概要

ファクトリーオートメーションやプロセスオートメーションにおける機械の監視制御を担う制御システムは，制御システムは，コンピュータ技術の発展にあわせて進化しており，制御システムコンポーネントの IoT 化や，IT と OT の連携が進むことで，多くのセキュリティホールを生むこととなった．そのセキュリティリスクの最小化は社会的な課題の一つである．制御システムにおいてセキュリティ対策を実践することは，制御システム固有の専門知識に加え，最新のサイバー攻撃のトレンドや対策技術といったセキュリティに関する知見が求められる．特にセキュリティ対策の立案時において，コスト対効果が高い制御システムのセキュリティ対策仕様を得るには，脅威モデリングの実施が求められるが，その実践コストは高く，制御システムのセキュリティ対策が進まない一因となっている．この課題に対し，制御システムの管理責任者が，脅威モデリングを実施するにあたり，脅威モデリングに関する知識の再利用性向上や自動化により効率化し，そのプロセスの属人性排除と，設計コストの低減を支援するシステムを実現することが，本論の目的である．特に本論では，脅威モデリングに係る要素のオントロジに基づき数理的にモデル表現することで，アプローチを試みる「オントロジ駆動型モデリング」に基づく脅威モデリング手法を提案した．脅威モデリングは大きく，リスク識別，リスク評価，リスク対処のプロセスから成る．各過程における既存手法で解決できない課題に対し，オントロジ駆動型モデリングに基づくモデル化手法，アルゴリズムを設計し，各過程における課題の解決方法を提案した．

リスク識別過程では，脅威シナリオの網羅性に着眼した先行研究に対し，その中でも優先的な脅威シナリオを得るための攻撃者のモデル化手法を開発した．リスク評価過程では，攻撃者を主体においた脅威シナリオの内部構造に基づき，従来の脆弱性情報のみを対象とした手法と比べ，リスクをより定量的に表現するリスク評価モデルを開発した．最後に，リスク対処過程では，コスト情報といったセキュリティ対策モデルが未知であっても，優先的なセキュリティ対策仕様を機械的に得る手法を開発した．

脅威モデリング各過程に対し，上記の提案手法を適用した結果，各過程の課題に対し，既存手法で解決できない課題を解決できる見通しを示した．そして，本論の目的に対し，脅威モデリングに関する知識の再利用性向上や自動化の実現するフレームワークをオントロジ駆動型モデリングにより実現できる可能性を示した．

目次

第1章 序論

1.1. 背景

Factory Automation (FA)や Process Automation (PA)における機械の監視制御を担う産業制御システム（以後，制御システム）は，制御システムは，コンピュータ技術の発展にあわせて進化している[1]．離散系システムの制御は古くはリレーで構築されていたが，1970 年代にマイクロコンピュータを採用する Programmable Logic Controller (PLC)[2]が登場し，飛躍的に制御ロジック開発効率が向上した．また，プラント産業で見られる連続系システムの制御においても，1960 年代にミニコンピュータが登場したことで，多変数の複雑な計算による高度制御を実現した．また 1970 年代からコンピュータネットワーク技術の発展とコンピュータの小型化が進み，Distributed Control System (DCS)[3]が登場し，システム構成の多様化が進んだ．

1990 年代になると，地理的に分散した各拠点のシステムを統合監視する Supervisory Control And Data Acquisition (SCADA)システム[4]を中心に Microsoft Windows™（以後，Windows）や Linux™（以後，Linux）ベースのプラットフォーム，TCP/IP などのエンタープライズシステムにおける標準技術の制御システムへの採用が進み，情報システムで確立した様々なシステム開発手法や運用ナレッジを取り入れることで，システム開発期間の短縮化や，制御システム部品のマルチベンダー化，Manufacturing Execution System (MES)，Manufacturing Operations Management (MOM)システムなどの業務系システムとの連携が進んだ[5]．2000 年代以降は制御システムのスマート化が進み，データ分析技術の進化や，Internet of Things (IoT)の活用により，制御システムの運用事業者にとって，製造工程の省エネ化，製造業務の省人化，生産性の向上，新たな事業価値の創生などの恩恵をもたらした[6]．Figure 1.1 に今日の典型的な制御システムの構成を示す[7]．

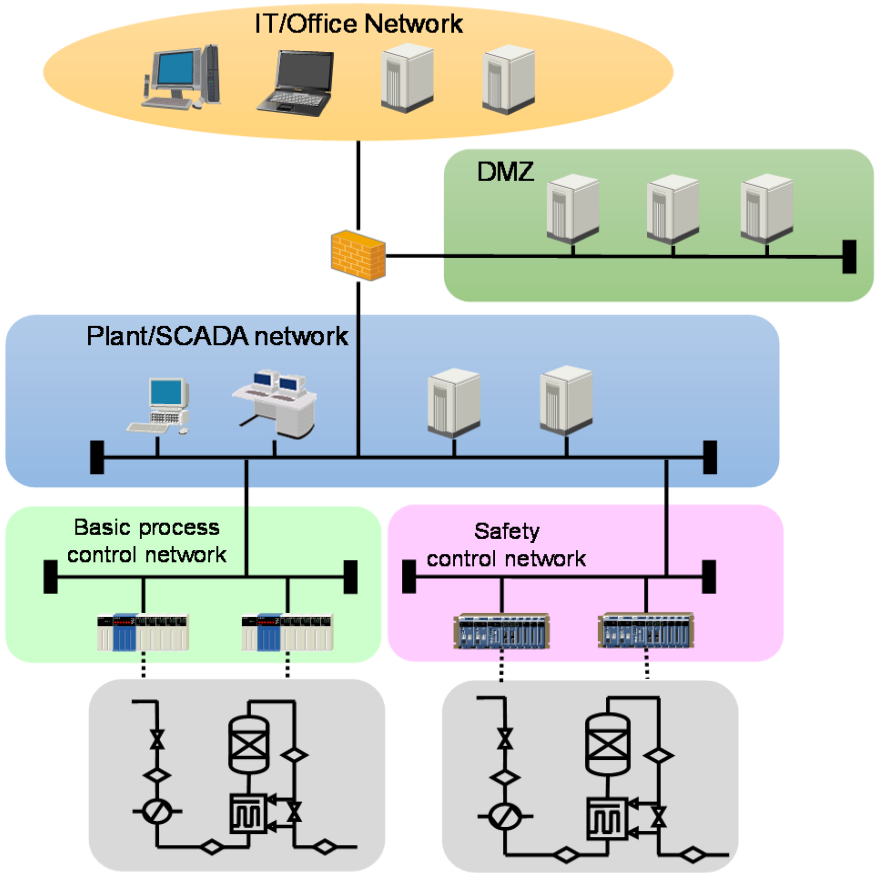


Figure 1.1　A typical structure of industrial control system

この構成は，Purdue model[8]に基づく典型的な制御システムのパターンである[9]．センサやアクチュエータから成るフィールドデバイス，フィールドデバイスを制御する制御器から成る制御系ネットワーク，各サイトの統合監視制御のためのシステムから成るプラント系（SCADA)ネットワーク，そしてエンタープライズ系業務システムからなる IT 系（Office)ネットワークから成る．IT 系ネットワークとプラント系ネットワークの間は，Demilitarized Zone (DMZ)が置かれ，このネットワーク上のサーバを介し，データ連携する．この構成により，IT 系とプラント系システムのデータ連携を可能にしている．

一方，制御システムを標的とするサイバー攻撃やインシデント報告が急増している． 2000 年代から，Conficker[10]や Ramnit[11]などのコンピュータウイルスによる感染が原子力発電所を始めとする多くの制御システムも影響を受けた．これらは制御システムの破壊を意図したものではなかった．ところが，2010 年には意図的に制御システムを破壊することを意図したマルウェアである Stuxnet[12]が初めて発見された． Stuxnet の発見以降， HAVEX[13] や Shamoon[14]， BlackEnergy3[15]， CrashOverride/Industroyor[16]，TRITON[17]といった制御システムを意図的に狙ったマルウェアや，Dragonfly[18]，APT33[19]といった制御システムを標的とする脅威アクターが次々と明らかになった．また，2020 年代以降は，ランサムウェアが制御システムを標的にし始め，制御システムの運用事業者からの金銭搾取を目的に攻撃を受ける事例が増えている[20]．制御システムの主なサイバー攻撃インシデントの事例を Figure 1.2，Table 1.1 に各インシデントの概要を示す．

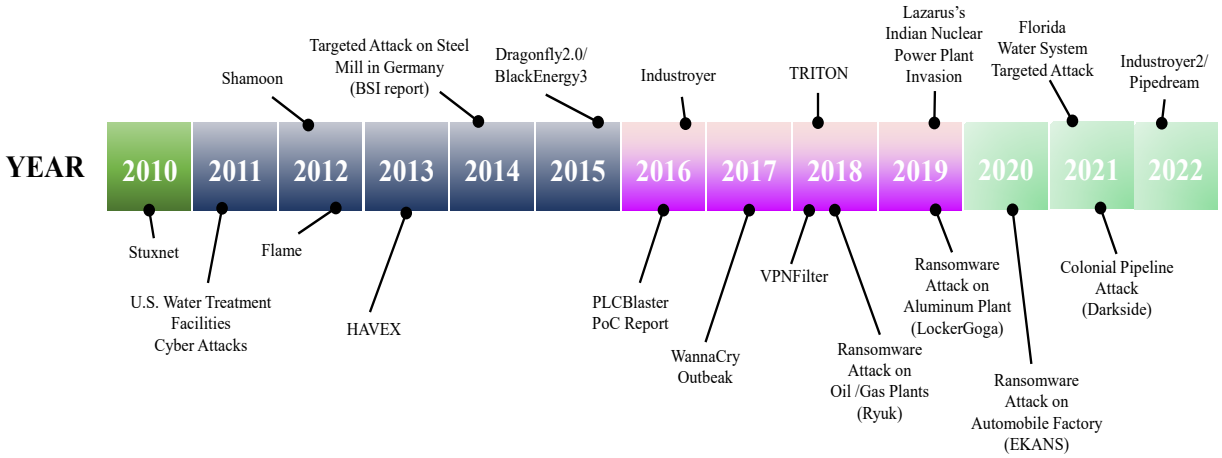


Figure 1.2 Examples of major cyberattacks on industrial control systems

Table 1.1 Description of major cyberattacks on industrial control systems

#	Incident Name	Year	Description
1	Stuxnet[12]	2010	Sophisticated malware targeting Iran's nuclear power plant
2	US. Water Treatment Facilities Cyber Attack[21]	2011	First reported cyber-attack damaging a U.S. water plant's operational system
3	Shamoon[14]	2012	Destructive malware targeting energy sector linked to Saudi Aramco attack
4	Flame[22]	2012	Sophisticated cyber espionage tool designed for data theft and surveillance with extensive spying capabilities, targeting middle eastern oil companies
5	HAVEX[13]	2013	Industrial espionage malware targeting energy sector via supply chain
6	German steel mill attack[23]	2014	Cyberattack causes physical damage to German steel mill
7	BlackEnergy3[15]	2015	Malware for cyber espionage and energy ICS attacks
8	PLC-Blaster[24]	2016	proof-of-concept malware self-spreading targeting Siemens S7 PLCs
9	Industroyer[16]	2016	Malware targeting industrial control systems, caused Ukraine blackout
10	WannaCry[25]	2017	Huge ransomware attack impacted on many ICS
11	VPNFilter[26]	2018	Malware targeting VPN devices, capable of espionage and destruction
12	TRITON[17]	2018	Malware targeting safety systems of industrial facilities
13	Ryuk[27]	2018	Ransomware attack impacted on oil/gas plant
14	Indian Nuclear Plant invasion[28]	2019	Cyberattack on Indian nuclear plant, suspected state-sponsored espionage.
15	LockerGoga[29]	2019	Ransomware targeting industrial and corporate networks including aluminum plant in Finland
16	EKANS[30]	2020	Malware targeting industrial control systems, disrupts processes, demands ransom.
17	Florida water system attack[31]	2021	Attempted cyber intrusion to poison Florida city's water supply by increasing sodium hydroxide levels remotely.
18	Colonial Pipleline attack[32]	2021	Ransomware attack on Colonial Pipeline, disrupted US fuel supply.
19	Industroyor2[33]	2022	Cyberattack with multiple malware to disrupt energy control systems

制御システムがサイバー攻撃を受けると，システム稼働停止に伴う生産停止，取引先との信頼関係の喪失，インシデント対応に係るコストといった金銭面の影響だけでなく，制御システムの制御対象によっては，健康や安全への影響，工場やプラント周囲の環境に対する悪影響に繋がる恐れがある．例えば，2010 年に発生した Stuxnet は，イランの原子力発電所内の何千台の遠心分離機の破壊に成功している[12]．他にも，2015 年と 2016 年に Dragonfly によるウクライナの電力系統に対するサイバー攻撃により，大規模停電に繋がった事例も報告されている[34]．

実被害に至らなかったものの，中東の石油化学プラントの制御システムの安全計装システムに対する攻撃[17]や，米国フロリダ州の上水システムに対する水酸化ナトリウム濃度を不正に上昇させるサイバー攻撃[31]も報告されている．また，ランサムウェアによる攻撃事例として，WannaCry を始めとする大きく感染が広まったランサムウェアによって，ERP や MES，SCADA などの制御システムに関するシステムが利用不能となり事業停止に繋がる事例が相次いだ．それらの中で特に大きなインシデントは，米国 Colonial pipeline が，Darkside によるランサムウェア攻撃の被害を受け，石油製品の供給が一時的に停止した事例[32]である．また，国内自動車製造業者の海外拠点を攻撃対象とする EKANS[30]と呼ばれるランサムウェアから，意図的に制御システムコンポーネントを狙った攻撃コードが発見された事例もある．

以上の例からわかる通り，制御システムに対するサイバー攻撃は，金銭的，社会的悪影響に繋がる恐れがある．したがって，制御システムのサイバーセキュリティの確保は大きな社会課題の一つである．

この社会課題に対して，制御システムのサイバーセキュリティを確保することを目的とする学術研究が多く報告されている．その研究範囲は多面的であり，主に制御システムを対象とするセキュリティ脅威分析，リスク管理フレームワークに関する研究[35][36][37][38]，制御システムの脅威検知に関する研究[39][40][41][42]，制御システムに特化したサイバー攻撃手法に関する研究[24][43][44][45][46][48]，制御システムに特化した防御手法に関する研究[47][49][50][51]がある．セキュリティ脅威分析に関する研究では，攻撃ツリー分析などの脅威分析手法の効率化やセキュリティリスク管理を自動化するフレームワークの確立を目的とした研究が報告されている．脅威検知に関する研究では，制御システムに特化した脅威検知のためのフレームワークや，制御システム向けに影響を与えない侵入検知手法に関する報告がされている．攻撃手法，防御手法に関する研究では，SCADA システムや制御システムネットワーク，PLC 等の制御システムコンポーネント，制御ロジックに対する攻撃手法や，それらをセキュアにするための関する手法に関する研究報告がされている．

1.2. 制御システムセキュリティに関する動向

制御システムをセキュアにするにあたり，無計画に様々なセキュリティ対策を取り入れるのでは，セキュリティ対策の効果を最大化することができない．セキュリティ対策の効果を最大化するには，安全設計や信頼性設計と同様，何らかの手法により，想定されるケースにおいて制御システムの性能劣化が発生しないように考慮した設計が必要である．制御システムは，エンタープライズシステムとはシステム構成や運用形態が大きく異なり，エンタープライズシステム向けの推奨対策をそのまま適用することはできない．Table 2.1 にエンタープライズシステムと制御システムとの違いを示す．

Table 2.1　Differences between enterprise systems and industrial control systems

Categories	Enterprise Systems	Industrial Control Systems
Computing Platform	PC/Workstation/ Server computer/ Cloud	PC/Workstation/ Server computer/ Embedded device
Priority of protected asset	Confidential information	Health, Safety, and Environment (HSE)/ Business Continuity
Responsible Department	IT Dept.	Production and Manufacturing Engineering Dept.
Application	Scalable functions	Fixed functions/ Cyclic processing
Life cycle	3-5 yrs	Over 10 yrs

処理基盤について，制御システムは制御処理や Remote Terminal Unit (RTU)等のフィールド機器などのゲートウェイなどの組込みデバイスが含まれる．保護対象の優先順位についても，情報の機密性(Confidentiality)保護が優先されるエンタープライズシステムに対し，制御システムは現場の健康安全や環境(HSE)，事業継続性(Availability)保護を優先するシステムが多い．システムの管理部署もエンタープライズシステムは IT 部門であるが，制御システムは製造部門や生産技術部門である場合が多い．システム上で動作するアプリケーションとして，エンタープライズシステムはスケーラブルなアプリケーションが多くみられるが，制御システムの場合，固定の機能で周期処理していることが多い．ライフサイクルについても，エンタープライズシステムは概ね 3 年から 5 年単位でシステムが一新されるが，制御システムの場合は 10 年以上稼働することも珍しくない．したがって，制御システムに適したセキュリティ対策を検討する必要がある．

制御システムに適用できるセキュリティ対策は技術対策からプロセス対策まで多様であり，多くの標準で推奨対策が示されている．例えば，NIST SP800-82 や IEC 62443 シリーズ[67]などでは，アクセス制御や暗号化，認証といった技術的な手段や，インシデント対応，セキュリティ監視，リムーバブルメディアの管理などのプロセス対策が制御システム向けに推奨対策として挙げられる．

制御システムのセキュリティを実践するには，その実践する主体を明らかにする必要がある．制御システムのセキュリティに関する国際標準である IEC 62443 における IACS (Industrial Automation and Control System)における役割モデルを Figure 2.1 に示す．IACS とは，制御システム実体だけでなく，それを管理運用する人，組織などの母体を含む概念である．このモデルでは，制御システムのステークホルダと，制御システムを構成する機能に対する役割を定義するモデルを定義している．このモデルは，セキュリティ対策は制御システムの管理し，運用する事業者 (Asset Owner，以後，制御システムオーナ) は，制御システム(IACS)に対する管理責任と運用の役割を有する．保守サービスプロバイダ(Maintenance Service Provider)は，制御システムの制御システム実体(Automation solution)と，保守プロセスに対して責任を持つ．制御システムインテグレータ(Integrated Service Provider)は，制御システムの構築に関して責任を持つ．更に，制御システムから独立した環境として，制御システム部品（製品）の開発と維持管理の責任を持つ製品サプライヤ(Product Supplier)も存在する．制御システムのセキュリティを検討する上

では，各ステークホルダが有する責任に基づき，必要とするセキュリティ上の責務を遂行する．例えば，制御システムオーナは，制御システムに係るセキュリティ管理上の問題すべてに対する最終責任を持ち，制御システムの運用面のセキュリティの実施主体となる．同様に，保守サービスプロバイダは，制御システム保守における技術面，および運用面のセキュリティ対策の実施主体なり，制御システムインテグレータは，制御システム開発時，試験時のセキュリティ対策の実施主体となる．制御システム各部品のセキュリティ対策は，各部品の製品サプライヤがその責任を持つ．

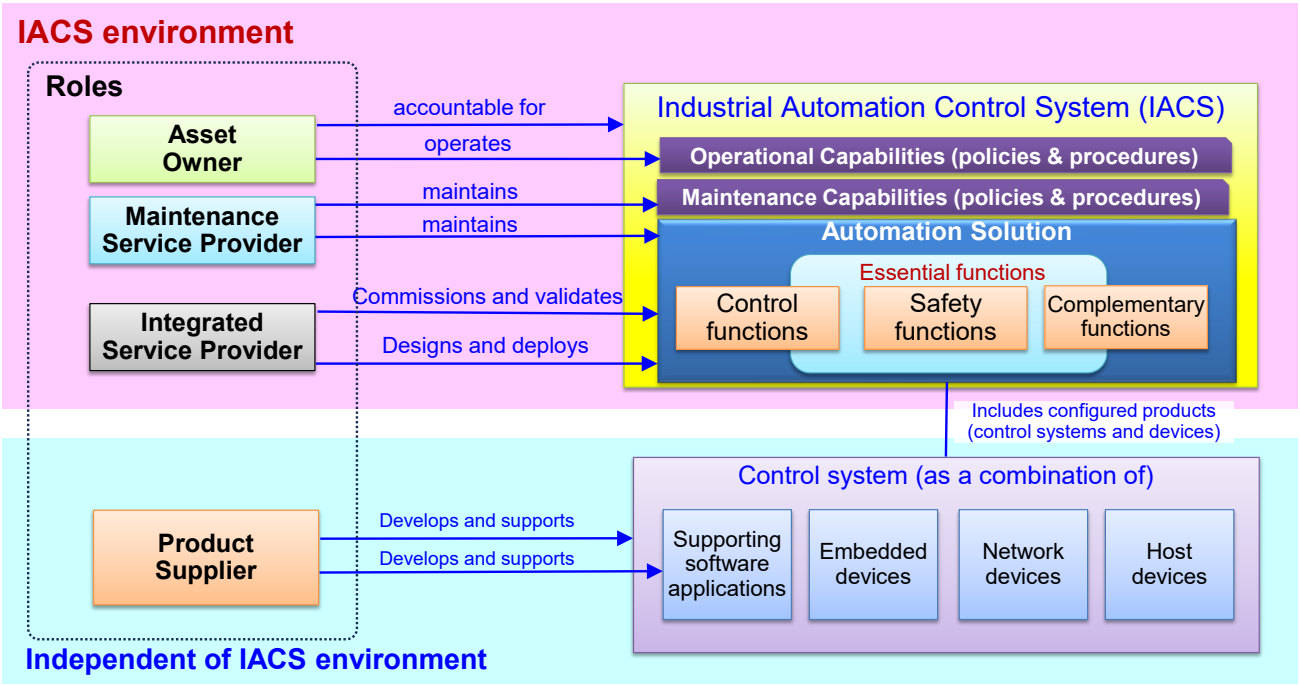


Figure 2.1　Roles and responsibilities model in IEC 62443

IEC 62443 では，ステークホルダ毎で推奨するセキュリティ標準化された対策が定義されている．最も簡単に制御システムのセキュリティを確保する方法は，標準に示されたセキュリティ対策を漏れなく実践することである．これはベースラインアプローチ[68]と呼ばれる．一方で，多くのセキュリティ対策は，導入コスト，または運用コストが少なからず発生する．制御システムオーナにとって，効果の薄いセキュリティ対策は余計なコストとなり，費用対効果の低いセキュリティ対策は導入すべきでない．

1.3.　本論の目的

制御システムのセキュリティリスクを完全に排除することは困難である．制御システムと外部ネットワークと隔離されており安全と言われていたが，Stuxnet により隔離されたネットワークであっても，侵入が可能であることが実証された．つまり，外部と何らかのデータのやり取りをしている制御システムは，少なからずサイバー攻撃を受けるリスクを有する．制御システムのスマート化に伴い，制御システムコンポーネントの IoT 化や，IT と制御システムとの連携が進むと，多くのセキュリティホールを生むこととなり，そのリスクの最小化が課題である．

制御システムにおいてセキュリティ対策を実践することは，最新のサイバー攻撃のトレンドや対策技術といったセキュリティに関する知見が求められる．制御システムの現場は，機械制御や電気設備等に関する装置やシステムも存在し，制御システム固有の専門知識も必要である．制御システムオーナにとって対策の立案や実践時に金銭面または人的コストが少なからず発生する．

セキュリティ対策に必要なコストは，セキュリティリスク分析，計画に係る「設計コスト」，対策の導入，セキュリティ機器導入やエンジニアリングなどの「導入コスト」，そしてセキュリティの継続監視や対処に係る「運用コスト」がある．セキュリティ対策の実務担当者にそれらの知見がなければ，コスト対効果（あるセキュリティ対策の導入コスト，運用コストに対する脅威への対処能力）が高いセキュリティ対策の見極めは容易でなく，費用対効果の高い制御システムのセキュリティ仕様を得るには，知識を有する専門家に委ねる必要がある．これが設計コストである．セキュリティの専門家が不足する多くの制御システムオーナにおいて設計コストは高く，制御システムのセキュリティ対策が進まない一因となっている．この課題に対し，工学的アプローチを適用し，コスト対効果の高いセキュリティ対策を得るプロセスを，脅威モデリング[52]と呼ぶ．制御システムオーナが，脅威モデリングを実施するにあたり，脅威モデリングに関する知識の再利用性向上や自動化により効率化し，そのプロセスの属人性排除と，設計コストの低減を支援するシステムを実現することが，本論の大目的である．

脅威モデリングは大きく，セキュリティ脅威を識別するプロセス（リスク識別），識別した脅威を評価し，対処すべき脅威を選定するプロセス（リスク評価），対処すべき脅威を基に，対処策を立案するプロセス（リスク対処）から成る．

リスク識別においては，制御システムに攻撃を仕掛ける攻撃シナリオ（攻撃ツリー，攻撃グラフ）を既存モデルの統合やコンテキストに注目して自動生成する先行研究が報告されている[53][54][55][56]．これらの既存手法は網羅的な脅威シナリオの自動抽出を目的とするが，優先順位付けの観点で課題がある．シナリオが網羅であるということは，その中から真に優先すべきシナリオの選定が困難になる恐れがある．リスク識別の観点で解決すべき本論の課題は，脅威モデルの再利用性を促進しつつ，優先的な脅威シナリオ・攻撃シナリオを抽出することである．

リスク評価においては，優先的な脅威シナリオ・攻撃シナリオを更に絞り込むために，リスクモデルの定量表現が必要である．既存研究では，リスクの定量表現の手段として標準化された指標を基にした定量化手法を提案されている[57][58][59]が，この手法は攻撃対象の脆弱性のみを対象としており，攻撃者自身の要素が考慮されていない．リスク評価の観点で解決すべき本論の課題は，攻撃者を主体においた脅威シナリオの内部構造に基づき，定量的なリスク評価モデルを提案することである．

また，リスク対処においては，先行研究では，攻撃ツリーや攻撃グラフといった可視化された攻撃シナリオに対して，既知のセキュリティ対策モデルを基に最善の対策を獲得する手法が報告されている[60][61][62][63]が，これは，コスト情報といった利用可能なセキュリティ対策モデルが既知である必要があり，そのモデルが不定の場合は適用できない．対策モデルを事前に得ることは難しく，セキュリティ対策モデルが不定であっても，優先すべきセキュリティ対策を決定できることが望ましい．

前述の通り，制御システムオーナは自社が所有する制御システムのセキュリティ確保に関して最終的な責任を持たなければならない．すなわち，脅威モデリングは制御システムオーナの責任で実施する必要がある．本論の大目的は，制御システムオーナが，脅威モデリングを実施するにあたり，脅威モデリングに関する知識の再利用性向上や自動化により，そのプロセスの属人性排除，および設計コスト低減を

支援するシステムを実現することである．この問題に取り組むにあたり，オントロジに基づき，脅威モデリングの各プロセスの実施に係る要素を数理的なモデルとして表現し，そのモデルを基に，上記課題の解決を試みた．この手法を本論では「オントロジ駆動型モデリング」と呼ぶ．この手法の詳細は次章にて詳説する．

1.4. 本論の構成

本論文の章構成を Figure 1.3 に示す．本論は全六章から成る．

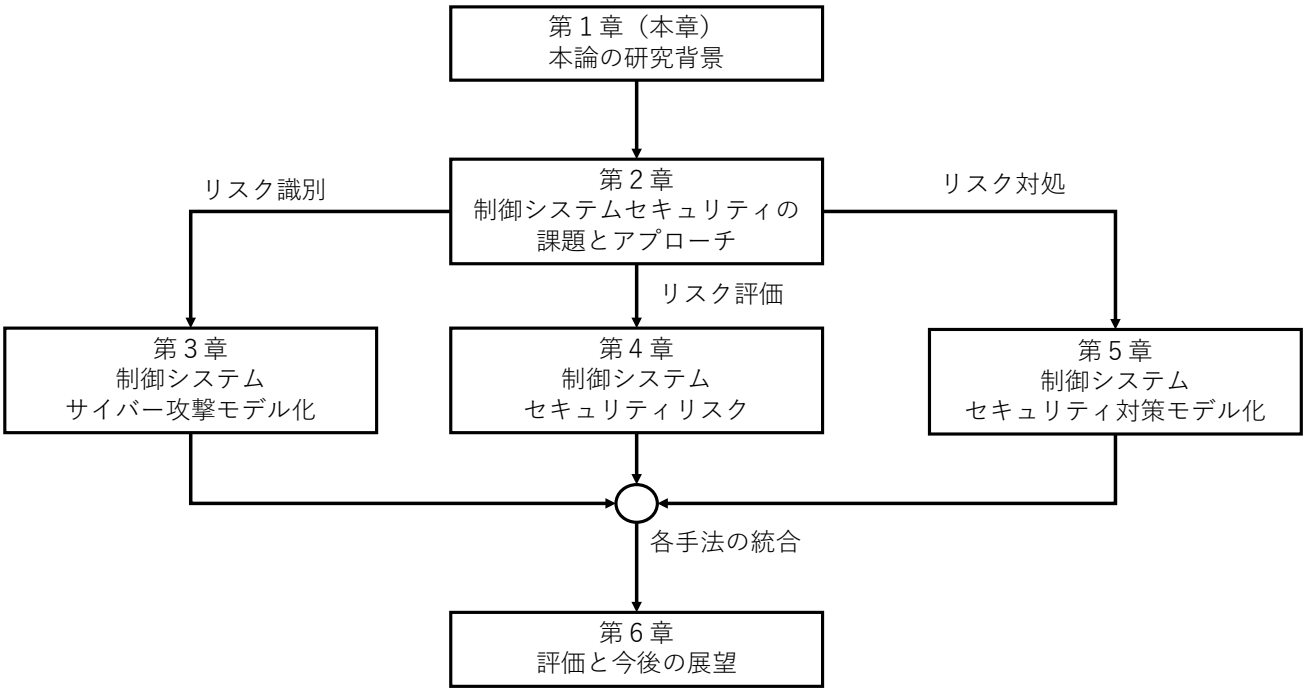


Figure 1.3 The chapter structure of this dissertation

第 2 章では，本論に関する議論の準備として，本論が対象とするセキュリティエンジニアリングプロセスと脅威モデリング，および本論で提唱するオントロジ駆動型モデリング，および関連する概念に関して簡潔に述べる．

第 3 章は，脅威モデリングにおけるリスク識別として，モデルの再利用性を担保する脅威アクターの状態モデルに基づく脅威シナリオの表現手法を詳説する．

第 4 章は，脅威モデリングにおけるリスク評価として，属人性を排除しつつセキュリティリスクの見積もる手段として，前述の脅威シナリオ表現モデルに基づくリスク計算モデルを詳説する．

第 5 章は，脅威モデリングにおけるリスク対処として，高リスクの脅威シナリオに対し，セキュリティ対策の前提知識不要で優先的なセキュリティ対策を実践すべき箇所を機械的に決定する手法を詳説する．

第 6 章は，各提案手法に基づき，本論で提案するオントロジ駆動型モデリングの制御システムのセキュリティエンジニアリングに対する有用性の評価を述べる．本章の後半で，制御システムのセキュリティエンジニアリングにおける今後の展望を述べる．

第2章 準備

2.1. セキュリティエンジニアリングと脅威モデリング

セキュリティエンジニアリングとは，セキュリティ対策の効果を最大化するために，システムライフサイクルの早期段階でセキュリティ保護要件，必要機能を定義し，システム開発を進めるアプローチである[64]．これは，セキュリティ要求工学[65][66]とも呼ばれる．セキュリティエンジニアリングを制御システムに適用することで，セキュリティ脅威に晒されている環境であっても，高信頼な制御システムを構築することができる．制御システムのセキュリティエンジニアリングにおいて，コスト対効果の高いセキュリティ対策を得るプロセスとして脅威モデリング[52]がある．脅威モデリングは，制御システムに起こりうるサイバー攻撃に起因する脅威事象を識別と，その優先順位を決定し，特に優先的に対処すべき脅威に対し，セキュリティ対策（検知，防御，または被害発生・拡大の抑止策）仕様を明らかにする手順である．脅威モデリングは，制御システムを効果的にセキュア化する手段として広く活用されている．例えば，制御システムにおけるセキュリティ設計とリスク評価の標準である IEC 62443-3-2:2020[69]は，Figure 2.21 に示す手順を推奨とする．IEC 62443-3-2:2020 は，最初に対象とする制御システム(SUG)を決定する(ZCR 1)．決定後，初期のリスク評価を実施し(ZCR 2)，制御システムをゾーンとコンジットに分割する(ZCR 3)．その後，各ゾーンのリスクを比較し(ZCR 4)，リスクの高いゾーンに対して詳細リスク評価を実施する(ZCR 5)．そして，ゾーンのリスク評価結果に基づき，セキュリティ対策の要求事項，仮定，制約条件を文書化する(ZCR 6)．

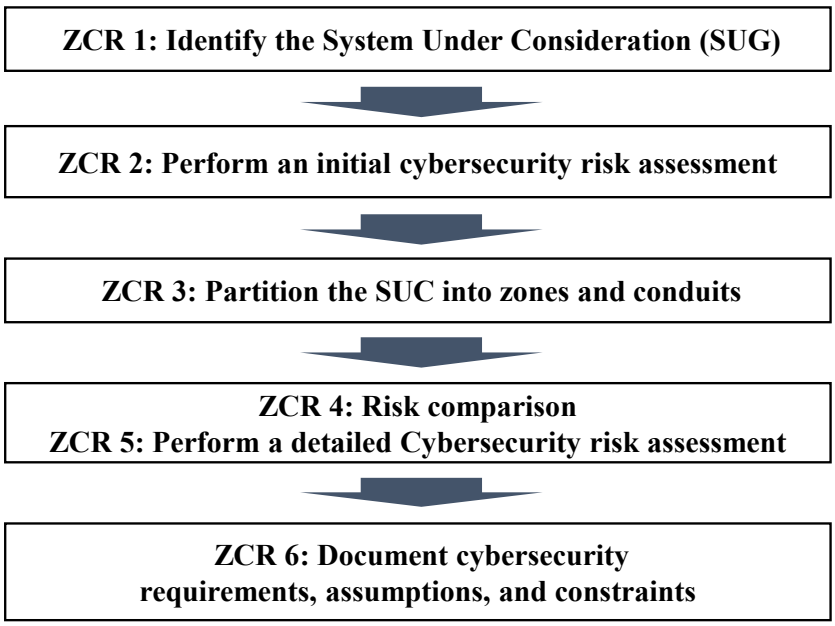


Figure 2.2　Summarized risk assessment process based on IEC 62443-3-2:2020

脅威モデリングの基本プロセスは，セキュリティ確保対象の制御システムに対して，リスクを識別（リスク識別）し，識別したリスクを評価（リスク評価）し，高リスクの事象に対して対策を立案する（リス

ク対処)．IEC 62443-3-2:2020 は，階層的に脅威モデリングを実施することで，高リスクのゾーンに対して優先的にセキュリティ対策を講じるアプローチを採用している．

リスク識別過程では，制御システムにおけるサイバー攻撃起因による脅威事象と，一連の脅威事象の発生に繋がる様々な攻撃者の一連の攻撃（攻撃シナリオ）を明らかにする．サイバーセキュリティの文脈において「脅威」と「攻撃」は観点が異なる．脅威とは，システムやシステムを所有する組織に対する被害をもたらす可能性のある，望ましくない事象である[70]．脅威には，意図的な事象と偶発的な事象の両方を含む．一方で，攻撃とは，脅威を実現する手段であり，セキュリティ・メカニズムを迂回や脆弱性を悪用により，システムに害を与えようとする意図的な試みである．本論では，制御システムを脅かす様々なタイプの攻撃者（脅威アクター）による一連の攻撃を「攻撃シナリオ」と呼び，攻撃の成功により起こる悪影響を含む一連のシナリオを「脅威シナリオ」と呼ぶ．

リスク評価過程では，脅威シナリオのリスクレベルを評価する．リスクレベルとは，そのリスクが実際に現実化した場合に及ぼす潜在的な悪影響の大きさと，その起こりやすさの両方を合わせた度合いを意味する．リスクレベルの評価は，攻撃シナリオがどれだけ容易に達成可能か，そして脅威が現実化した場合に予想される悪影響の大きさにより決定される．例えば，攻撃の技術的な難易度や攻撃に必要なリソース，システムのセキュリティ対策，および脅威が現実化した結果に基づき評価する．リスクレベルが高い脅威シナリオは，攻撃が容易に達成可能であり，脅威が現実化した結果として発生する悪影響が大きいことを意味する．

リスク対処過程では，脅威シナリオに対してセキュリティ対策仕様を決定する．この段階では，リスクの低減だけでなく，リスクの受容，転嫁，または回避する戦略を検討する[71]．リスクに対処するセキュリティ対策の例として，ファイアウォールの設置やアクセス制御といった，システムに適用する技術的対策や，アクセス権管理やセキュリティ監視の強化など運用的対策，要員に対する教育訓練の実施，セキュリティポリシの策定と周知(セキュリティマネジメントシステム)といった管理的対策などがある[7]．

脅威モデリングは，システム構成情報があれば実施可能である一方，その実践は，セキュリティに関する基礎知識，制御システムに対する脆弱性や攻撃手法やセキュリティ対策に関する知識が不可欠である．また，脅威モデリングに係る工数も課題である．脅威シナリオは，攻撃手法や攻撃経路，脅威となりうる事象などの要素があり，それらの組み合わせで表現されることから，一つのシステムに対して多数の脅威シナリオが考えられる．すなわち，制御システムが大規模かつ複雑な場合は，様々な脅威シナリオのパターンが考えられる．そのため，脅威モデリングの実践は，経験を有する専門家チームが時間をかけて対応する必要があり，システム構成情報の収集過程も含め，その準備と実施に人的リソースが必要となる．

一つの工場やプラントを見ても，様々なシステムが稼働しており，限られたリソースですべてのシステムに対して脅威モデリングを実施することは困難である．脅威モデリングを効率化する手法として，STRIDE[72]や DREAD[73]，PASTA[74]などの脅威モデルが提唱されているが，それらモデルを適用しても尚，人手による作業の完全排除をできず，人手で可能性のある攻撃手法や一連の脅威シナリオの攻撃ツリーを作成していることが現状である．人手による作業を排除するには，脅威モデリングの実施に関する知識を個々のケースにあわせて再利用するアプローチが有用である．

本研究で解決すべき課題は，制御システムに対する脅威モデリングにおいて，知識の再利用性確保し，自動化を実現することである．この課題を解決することは，脅威モデルの知識表現の深化によるセキュリティ知識工学における新たな学際的枠組みの開拓に繋がる可能性を秘めている．

本論では，これの課題を解決するオントロジ駆動の概念を適用した方法論（オントロジ駆動型モデリング）を提案する．

2.2. モデリングの概要

モデリングは，知識やシステム，事象の構造を論理的な方法で示す手法であり，多くの科学技術分野において必要不可欠なプロセスである．制御システムに関するモデリングとしては，制御対象（物理現象，化学反応など）と，その入出力や状態を数学的なモデルとして表現するモデリング（制御工学的なモデリング），情報，データの内容と構造を可視化するモデリング，とシステムの構造や振る舞いを表現するモデリング（システム工学・ソフトウェア工学的なモデリング）がある．

制御工学的なモデリングとして，対象の原理・構造が完全に既知である場合に適用可能な第一原理モデリング，観測データを基にモデルを求めるシステム同定モデリング，部分的に既知の原理，構造を基にグレーボックスモデリングがある[75]．制御工学的なモデリングは，物理現象を直接制御することから，厳密なモデルが得られやすい点が特徴の一つである．

一方で，システム工学・ソフトウェア工学的なモデリングは，情報システムの設計において利用される．この文脈でのモデリングは，ソフトウェア開発における業務モデルの定義が主目的であり，特に複雑な業務のパターン化や業務の抽象モデルを定義する．代表的なモデリング手法（モデリング言語）として，ERD (Entity-Relation Diagram)[76]，UML (Unified Modeling Layer)[76]，SysML (Systems Modeling Language)[78]などがある．複雑なシステムを抽象化したモデルとして表現できる反面，制御工学的なモデリングと比べて厳密なモデルが得られにくい点が特徴である．

制御システムに係るセキュリティに関する事象は，制御システムのモデルが定義されていないと議論できない．これは，セキュリティを確保する対象の制御システムのモデルが明確でなければ，その脅威モデルや対策モデルといったセキュリティモデルを設計できないためである．また，「セキュリティ」自体が抽象的な概念であり，セキュリティという概念そのものに対して，実体モデルは定義できない．セキュリティ事象の仕様を記述するには，前述のセキュリティを確保する対象に対し，攻撃や対策の事例といった，実体が存在する必要がある．すなわち，セキュリティの文脈でモデル化する上での必要条件は，「セキュリティを確保する対象のモデルが明確であること」，および「セキュリティを確保する対象に対して，その実体が存在すること」と言える．制御システムにおいては，上述のモデリング手法により，業務システムから制御系に至る全ての階層でモデルが定義できることに加え，前章で述べた通り，多くのセキュリティ事例が存在することから，制御システムのセキュリティの文脈でモデル化することは可能であるといえる．

2.3. オントロジ駆動型モデリング

オントロジとは，知識を量子化された要素からなる数理的な概念として表現し，要素間の関係を構造的に表現するフレームワークである．オントロジ（存在論，以後オントロジで統一）は元々哲学由来であり，「存在」とは何かを論理的に紐解く学問である．情報科学の文脈におけるオントロジは，ある事象の概念と概念間の関連を数理的モデルとして表現することである[79]．オントロジは主に人工知能や知識工学の分野で応用されており，モデル化対象の構造や要素を明らかにし，コンピュータが理解，処理できるよ

うな数理的構造への変換に利用される[80].

オントロジ駆動型モデリングとは，本論ではオントロジに基づくモデリングアプローチと定義する．オントロジ駆動とは，オントロジに基づく要素を定義し，知識を再利用するフレームワークの意味を持つ[81]．オントロジは，情報システムやソフトウェア開発において，多く活用されている[81][82][83].

脅威モデリングは制御システムに対する様々なサイバー攻撃の内容，影響，防御手法といった知識に基づき実施する．制御システムにおける脅威モデリングに関する知識のオントロジを明らかにし，そのオントロジに基づきモデル化することで，脅威モデリングプロセスにおける知識の再利用性確保と自動化を期待できる．

2.4. アプローチ

上述の通り，脅威モデリングは大きくリスク識別，リスク評価，リスク対処の三つのステージから構成される．これら各過程にオントロジ駆動型モデリングを適用する．

最初に，リスク識別過程では，セキュリティ確保対象となる制御システムそのもの，制御システムにおいて想定される脅威アクター，脅威アクターの攻撃活動，および攻撃が成功した場合の被害をモデル化の対象とする．これらをモデル化し，脅威シナリオのモデル表現を試みた．本論では，脅威シナリオモデル化手法として，モデルの再利用性を担保する脅威アクターの状態モデルに基づく脅威シナリオの表現モデルを提案する．本提案の詳細は第 3 章にて述べる．

次に，リスク評価過程では，脅威シナリオのセキュリティリスクをモデル化する．リスクとは，「生起する事象の確からしさと，それによる負の結果の組合せ」である．つまり脅威シナリオが現実化する可能性と，脅威シナリオが現実化した結果として発生する影響の組み合わせが，脅威モデリングにおいて扱うリスクである．このリスクを決定するメカニズムをモデル化する．本論では，属人性を排除しつつセキュリティリスクの見積もる手段として，前述の脅威シナリオ表現モデルに基づくリスク計算モデルを提案する．本提案の詳細は第 4 章にて述べる．

最後にリスク対処過程では，脅威による被害の防止・対処策の決定する．リスク識別とリスク評価過程で明らかになる脅威シナリオに対し，効果的に対処する防止・対処策を決定するロジックをモデル化する．本論では，高リスクの脅威シナリオに対し，セキュリティ対策の前提知識不要で優先的なセキュリティ対策を実践すべき箇所を機械的に決定する手法を提案する．本提案の詳細は第 5 章にて述べる．

2.5. 提案手法の検討スコープ

工場やプラントにおいて稼働する制御システムは，情報処理システムに加えて，計器や制御対象などの物理的な機器（プラント）を包含する．本論におけるセキュリティ対策の検討範囲を Figure 2.32 に示す．

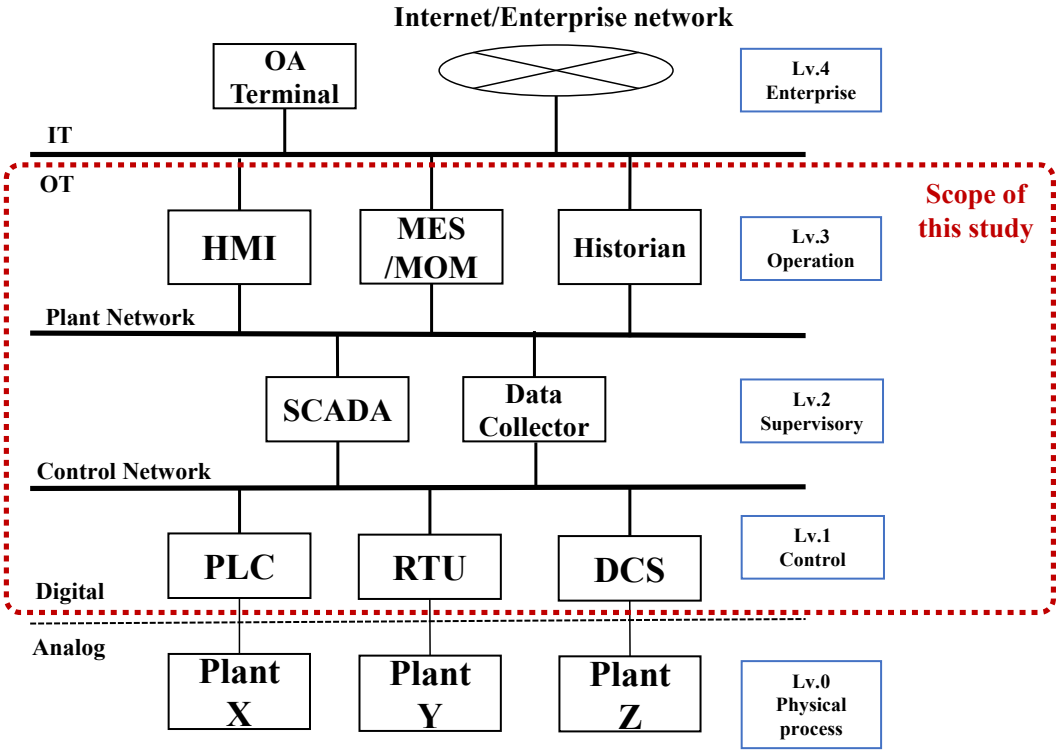


Figure 2.3　The scope to be secured for this study

本論においてセキュリティ確保する範囲は，PLC，DCS，RTU などのコントローラと，各コントローラからの情報収集や統合管理するための SCADA や Data collector 等が参加する制御系ネットワーク(Control network)と，HMI や MES/MOM，Historian などから成るプラント系情報システム(Plant network)，およびそれらネットワークに接続する各コンポーネントとする．

第3章 リスク識別：制御システムに対する攻撃活動のモデル化

3.1. はじめに

前章に述べた通り，脅威モデリングの最初のプロセスは脅威シナリオを明らかにすること，つまりリスク識別である．制御システムに想定される脅威事象（制御システム内外で発生する脅威に繋がる何らかの事象）と，脅威シナリオを識別するにあたり，脅威アクターが利用しうる攻撃手段や，その成立条件といった脅威に関する知識が必要不可欠である．

攻撃グラフや攻撃ツリーのような形式で脅威を識別し可視化する手法の事例は多く，特に攻撃グラフや攻撃ツリー自身は確立した手法である．一方で，攻撃を識別するために必要な知識量は個人差が大きく，脅威事象の識別結果が属人的となる傾向がある．上記知識のモデル化において，知識表現における属人性の排除は，本研究目的の達成上，解決すべき課題の一つである．この課題に対して，オントロジ駆動型モデリングに基づくアプローチを検討した．

オントロジは，形式表現された知識の共有と知識を必要とするプロセスの自動化手段として活用されており，属人性の排除を達成する方法の一つとして，情報伝達，システム間相互運用，システムエンジニアリングの分野を中心に有用性が認められている．オントロジを活用し，脅威事象を構成する要素と内部構造を明らかし，表現モデルを具現化することでリスク識別プロセスの効率化，自動化を期待できる．

先行研究において，オントロジに基づくリスク識別を自動化，効率化する手法の報告がある．例えば，脅威シナリオをオントロジに基づき分析し，モデル化する取り組みの報告がいくつかある[85][86]．一例として，自動化のためのフレームワークとして，例えば，制御システムに対して影響を与える攻撃目標に対して，当該事象を Cyber Kill Chain[87]，OWASP Attack Surfaces[88]，CAPEC (Common Attack Pattern Enumeration and Classification)[89]をはじめとする既存モデル間の関連を定義し，特定のイベントに連なる脅威を自動的に特定し，それに基づく攻撃ツリーを機械的に生成するフレームワークが提案されている[53]．これにより，モデルの再利用性を確保し，網羅的な攻撃ツリーを機械的に生成できる．他の先行研究では脅威を表現するコンテキストを元に脅威を構成する要素に分解し，それぞれについて脅威モデルをパターン化する手法の提案がある．この手法は要素の網羅性を担保することで，脅威事象を網羅的に識別するのと，再利用性を同時に確保する[54][55][56]．また，リスク識別を効率化する手法として，制御システムの資産モデルに基づき，リスク識別する報告もある[90]．

一方で，脅威事象の優先順位を決定する観点で課題が残る．多面的かつ網羅的に脅威シナリオを識別することは，考慮すべき脅威シナリオが増え，真に優先対処すべきシナリオを見逃す恐れがある．真に必要なのは，攻撃してくる可能性が高い脅威アクターにより引き起こされる高確率なシナリオを，網羅性を確保しつつ得ることである．これを踏まえ，攻撃してくる可能性が高い脅威アクターによって引き起こされる脅威モデルの再利用性を担保しつつ，かつ一貫性のある形式で表現し，個人の知識に依存しない方法で優先的に対処すべき脅威シナリオを識別することを可能とする，脅威事象を表現する構造化されたモデルのフレームワークを開発した．このモデルは，攻撃者の挙動と中間的な状態の構造を表現できるものである．本章では，この表現モデルの詳細について説明する．

3.2. 提案モデル

3.2.1 基本モデル

攻撃者は，攻撃の開始ポイントから最終目標とする結果に繋がるまで，制御システムに対してあらゆる攻撃を実行する．システムに侵入した攻撃者の挙動を分析するモデルとして Diamond model[91]がある．このモデルは Figure 3.1 に示された Adversary，Capability，Victim，および Infrastructure の四つの基本要素と，それら関係を表すモデルである．

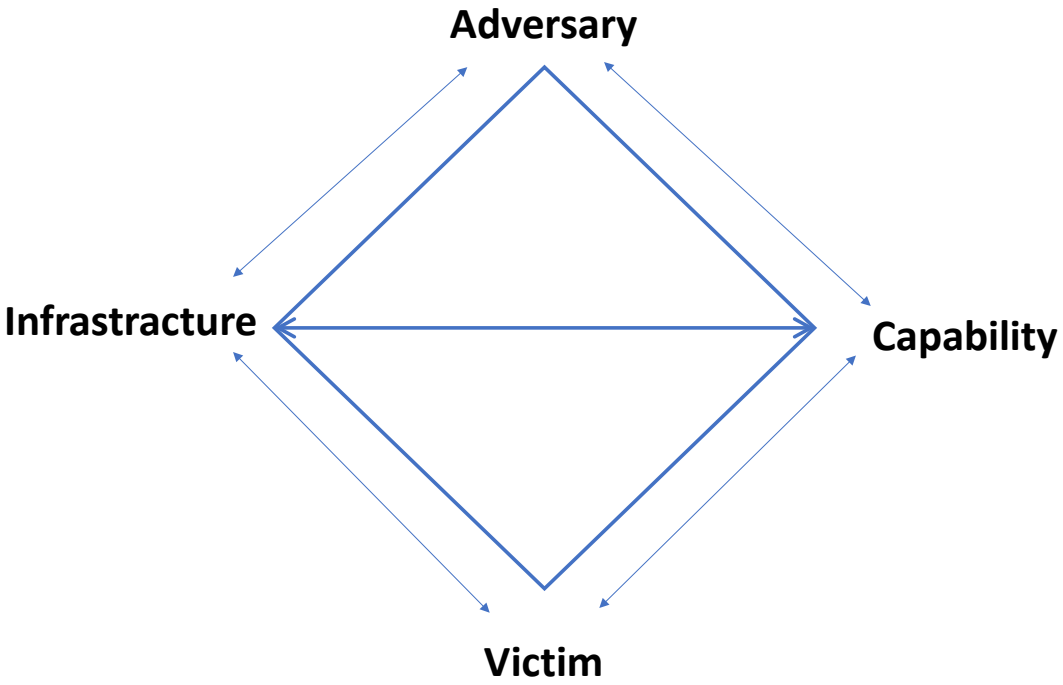


Figure 3.1　Diamond model

各要素の関係性を元に実際の攻撃事象の関係を明示的に表現する．Adversary は，システムに対して攻撃を試みる敵対者であり，攻撃を仕掛ける個人，グループを指す．Capability は攻撃者 (Adversary)が有する攻撃能力であり，攻撃者が実行可能な攻撃手法やツールを指す．Victim は，ある攻撃実行時点で対象とするシステムの要素を指す．Infrastructure は，攻撃者と攻撃対象の間に存在するシステム構造や制約を意味し，例えば，ネットワーク，ノードなどが該当する．

モデルの要素間を繋ぐ線は，攻撃行為で発生する事象間で起こる関連を意味する．これは，Adversary が，ある Victim 対して攻撃を実施するにあたり，Adversary は，Infrastructure，または Capability に該当する要素が，その攻撃の過程で存在することを意味する．Diamond model は，攻撃者がシステムに侵入した後の挙動を解析するために開発されたモデルであり，攻撃者自身や環境の前提や，攻撃の過程に対するモデル化に適する．

3.2.2 モデルの設計

Diamond model を基に，設計した攻撃対象とする制御システムの構造モデルと，攻撃者の振る舞いモデルを以下に示す．

A. 構造モデル

攻撃活動の構造を定義する．その構造は，制御システムに対して侵入試行する脅威アクター($ThreatActor$)，および攻撃の対象とする制御システム ($TargetSystem$)から構成されるモデルである．以下，モデルの制約を示す．

I. $TargetSystem$は，攻撃対象とする制御システムのモデルである．制御システムは，コンポーネント，コンポーネント間の論理的な接続関係を意味するネットワーク，およびコンポーネントが置かれる物理エリアから成る．(Condition .i)
II. $Victim$は$TargetSystem$内に存在する．更に，攻撃者の$Infrastructure$は$TargetSystem$の要素が含まれる．(Condition .ii)
III. $ThreatActor$は，何らかの攻撃目的($Goal$)を有する，特定の攻撃能力($Capability$)を有する単一，もしくは複数からなる敵対者($Adversary$)である．(Condition .iii)
IV. $Goal$のモデルは，$Victim$と攻撃活動の組とする．(Condition .iv)

以上の制約を踏まえ，定義したモデルを式(3.1)-(3.3)に示す．

$$TargetSystem = \{Component, PhysicalArea, Network\} \tag{3.1}$$

$$Victim \subseteq Component \tag{3.2}$$

$$\begin{aligned} Infrastructure = \{\, C_{inf}, P_{inf}, N_{inf} \mid C_{inf} \in Component \cup Component_{Adv}, \\ P_{inf} \in PhysicalArea \cup PhysicalArea_{Adv}, N_{inf} \in Network \cup Network_{Adv} \,\} \end{aligned} \tag{3.3}$$

式(3.1)は，Condition .i を指す．つまり，$TargetSystem$は，コンポーネント要素の集合から成る群，物理エリア要素の集合から成る群，ネットワーク要素の集合から成る群から成る集合族とする．この式では，$Component$は$TargetSystem$を構成するコンポーネントを要素として持つ有限集合とする．例として，$TargetSystem$のコンポーネント要素を$\mathrm{cp}_1, \mathrm{cp}_2, \mathrm{cp}_3, \ldots$とすると，$Component$は式(3.4)で表される．

$$Component = \{\mathrm{cp}_1, \mathrm{cp}_2, \mathrm{cp}_3, \ldots\} \tag{3.4}$$

同様に，$PhysicalArea$は，$TargetSystem$の物理エリア要素から成る有限集合である．$TargetSystem$の物理エリア要素を$\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \ldots$と仮定すると，式(3.5)で表される．

$$PhysicalArea = \{\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \ldots\} \tag{3.5}$$

$Network$は，$TargetSystem$のネットワーク要素から成る有限集合である．$TargetSystem$のネットワーク要素を$\mathrm{n}_1, \mathrm{n}_2, \mathrm{n}_3, \ldots$と仮定すると，式(3.6)で表される．

$$Network = \{n_1, n_2, n_3, \dots\} \tag{3.6}$$

式(3.2)(3.3)は，Condition .ii に関連する．式(3.2)が意味するのは，攻撃対象から成る集合$Victim$は，$Component$の部分集合であることである．式(3.3)に示す$Component_{Adv}$は，$TargetSystem$外の攻撃者の制御下にあるコンポーネントである．$Component_{Adv}$に属するコンポーネントの一例として，$TargetSystem$に元から存在しない攻撃者が利用するコンピュータが挙げられる．$PhysicalArea_{Adv}$は，$TargetSystem$に存在しない，攻撃者が利用できる物理エリアであり，同様に$Network_{Adv}$は，$TargetSystem$外の攻撃者が利用できるネットワークである．

$TargetSystem$は，構成に基づく制約がある．一例を挙げると，コンポーネントが 0 の$TargetSystem$は存在しない．このように，$TargetSystem$の構成要素は，それが存在しうるための制約条件を有する．事実に基づき，整理した$TargetSystem$と各要素間の関連を概念モデルで表現する．この関連モデルを ERD で表現した図を Figure 3.2 に示す．

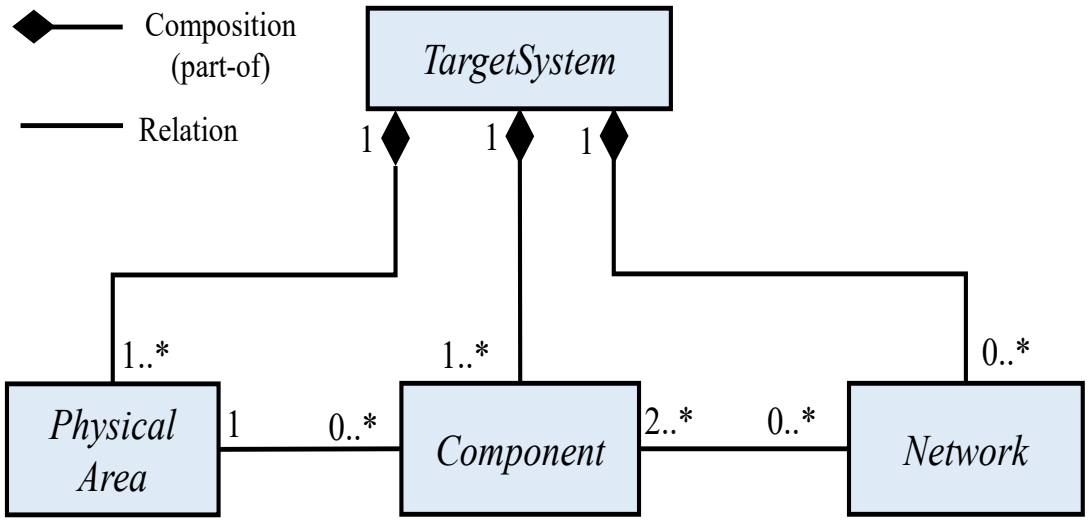


Figure 3.2　ERD of $TargetSystem$

$TargetSystem$は，1 以上の$Component$，1 以上の$PhysicalArea$，0 以上の$Network$から成る．$Component$は，1 箇所の$PhysicalArea$に存在する．逆に，$PhysicalArea$には 0 以上の$Component$が存在する．1 つの$Network$に 2 以上の$Component$が存在する．$Network$の最小単位は 2 台の$Component$からなるピア接続である．尚，$Component$が 1 の場合，$Network$は存在しえない．

次に，設計した脅威アクターモデルを詳説する．脅威アクターのモデル$ThreatActor$は，Diamond model における Adversary，Capability，および Adversary の目標(Goal)から成る(Condition .iii)．これより設計した$ThreatActor$モデルを式(3.7)で表す．

$$ThreatActor = (adv, Cp, G) \tag{3.7}$$

advは攻撃者実体であり，$adv \in Adversary$を満足する．ここで，攻撃者の実体集合を$adv_1, adv_2, adv_3, \ldots$とすると，$Adversary = \{adv_1, adv_2, adv_3, \ldots\}$として表現される．

Cpは，攻撃者が利用できる技術，ツール群である．ここで，技術やツールの実態を$tec_1, tec_2, tec_3, \ldots$とすると，$Cp = \{tec_1, tec_2, tec_3, \ldots\}$と表現される．

Gは攻撃者の最終目的の集合であり，複数存在し得ることから，最終目的の要素からなる集合を$Goal = \{g_1, g_2, g_3, \ldots\}$とした場合，$G \subset Goal$を満たす．

次に，$Goal$のモデルを定義する．$Goal$の要素は，$TargetSystem$上にあるコンポーネント(v_{last})，攻撃のアクション(act)から成る 2 組の組（直積集合）で表す(Condition.vi)．これらより，設計した$Goal$は式(3.8)となる．

$$Goal = \{(v_{last}, act) \mid v_{last} \in Component, act \in Cp\} \tag{3.8}$$

式(3.8)中のactは，Cpの要素である．これは,「攻撃のアクションを遂行する能力を攻撃者が有することは，攻撃者が目標を達成するうえで必要条件である」という定理と一致する．

次に，攻撃アクションが成功する条件を述べる．攻撃アクションが成功するためには，上述の通り，$Adversary$がactを実行できるだけの攻撃能力を有する必要がある．この定式化を試みる．ある$Adversary$が利用できる攻撃パターンからなる有限集合を，$Capability_{Adv}$とする．この場合，$Adversary$の攻撃アクション成功に必要な条件は式(3.9)となる．

$$act \in Capability_{Adv} \tag{3.9}$$

B. 振る舞いモデル

前節に示したモデルを元に攻撃者の振る舞いモデルを定義する．モデルの制約条件は以下である．

I. $ThreatActor$によって引き起こされる攻撃シナリオ($AttackScenario$)は，攻撃状態（ある攻撃が成功した直後より，次のアクションを遂行する迄の攻撃者の状態）から構成される．(Condition .v)
II. $ThreatActor$は，ある攻撃状態において，$ThreatActor$が有する能力の下，攻撃を遂行する．(Condition .vi)
III. ある攻撃状態に実行される攻撃パターンは，ターゲット，および攻撃フェーズに依存する条件（攻撃条件）がある．この条件は，攻撃者の攻撃位置，および攻撃時の条件から成る．(Condition .vii)
IV. 攻撃条件とは，ある攻撃状態の攻撃活動の成功可能性を高める条件の集合である．(Condition .viii)
V. 攻撃過程において，$Victim$は変化することがある．(Condition .xi)

上記の$AttackScenario$は，Condition .v より，攻撃状態からなる有向集合で表現される．$AttackState_k$を攻撃ステップ k の攻撃状態とした場合，$AttackScenario$は式(3.10)で表される．

$$AttackScenario = AttackState_1 \prec AttackState_2 \prec \cdots \prec AttackState_k \prec \cdots \prec AttackState_n \tag{3.11}$$

$A \prec B$は，順序関係を保持する有向集合であり，「Aは，Bの先行する」関係を表す．k，nは整数であり，$\{k, n \in \mathbb{N} \mid 0 \leq k \leq n\}$を満たす．$AttackState_1, \ldots, AttackState_n$は，$1, \ldots, n$番目の攻撃状態を示す．

次に(Condition .vi) ~ (Condition .xi）を元に設計した$AttackState_k$の定義を式(3.12)に示す．

$$AttackState_k = (p_k, v_k, Cp_k, lo_k, Cond_k) \tag{3.12}$$

式(3.12)の$AttackState_k$は変数の組で表され，k 番目の攻撃フェーズ（攻撃フェーズの基本要素からなる集合$AttackPhase$の要素 $p_k \in AttackPhase$），当攻撃フェーズにおけるターゲット$v_k \in Component$，攻撃フェーズ p_k，およびターゲットv_kに対する攻撃能力から成る有限集合$Cp_k \subset Capability_{p_k,v_k}$，$ThreatActor$の攻撃位置$lo_k \in TargetSystem$，および攻撃条件の有限集合$Cond_k \subset AttackCondition$ から成る．

式(3.12)において，$AttackState_k$は，k 番目の攻撃フェーズ $p_k | p_k \in AttackPhase$，該当攻撃フェーズにおける対象$v_k | v_k \in Component$，$v_k$に対する攻撃能力の集合$Cp_k | Cp_k \subset Capability_{p_k,v_k}$，$ThreatActor$の攻撃位置$lo_k | lo_k \in TargetSystem$，および$p_k$における攻撃条件の有限集合$Cond_k \subset AttackCondition$から成る変数の組である．ここで$AttackCondition$は(Condition .viii)に示す攻撃条件の集合である．これは，$TargetSystem$内に存在する全ての認証情報（クレデンシャル情報）を要素とする有限集合$Cred_{ALL}$，および$TargetSystem$内に存在するすべてのコンポーネントの操作権限を要素とする有限集合$Admin_{ALL}$から成る．式(3.13)に$AttackCondition$を示す．

$$AttackCondition = \{Cred_{ALL}, Admin_{ALL}\} \tag{3.13}$$

$ThreatActor$は，攻撃遂行時に上記に示した攻撃条件を悪用することで攻撃の成功確率を高める．これは，例として，あるコンポーネントの認証（クレデンシャル）情報が，他のコンポーネントの認証情報と同じ場合，一方のコンポーネントから搾取した認証情報を使いまわすことで認証を容易に突破することが可能となる．操作権限の場合も同じように働く．例えば，あるコンポーネント$\mathrm{cp_A}$が，別個のコンポーネント$\mathrm{cp_B}$の操作権限を所有する場合，$\mathrm{cp_A}$に対する操作権限が$ThreatActor$によって搾取された場合，$ThreatActor$は，$\mathrm{cp_B}$の操作権限も同時に獲得できてしまう．

上記を数理的に定義する．$AttackState_k$の到達時に$ThreatActor$が，cp1，cp2のクレデンシャル情報と操作権限を搾取したする．cp1，cp2それぞれのクレデンシャル情報を$(cred_1, cred_2)$とし，操作権限$(admin_1, admin_2)$とすると，$Cond_k$は式(3.14)となる．

$$Cond_k = \{cred_1, cred_2, admin_1, admin_2\} \tag{3.14}$$

ある攻撃状態から次の状態に遷移するには，攻撃者がその時点で利用できる攻撃技術が存在する必要がある．これは，任意のシステムに対して意図した状態を実現させたい場合に，その方法が無ければ，その状態を実現できないという公理[92]に基づく．すなわち，$AttackState_k$が$AttackState_{k+1}$に遷移する必要条件は，式(3.15)を満たす．

$$Cp_k \neq \phi \tag{3.15}$$

この式は，攻撃フェーズp_kに式(3.15)が成立しない場合，攻撃者は利用できる攻撃手段が無いことを説明する．つまり，$AttackScenario$はp_kで終了となる．

最後に，最終目標の到達性を説明する．$AttackScenario$において，式(3.8)の$Goal$達成時点における攻撃状態を，$AttackState_{END}$と置く．$AttackScenario$の最終目標への到達性は，$AttackState_{END}$に到達するか否かで決まる．整数kに対して，$ThreatActor$が$AttackState_k$から，$AttackState_{k+1}$に状態遷移する際，式(3.8)の$Goal$として定義された要素，つまりv_{last}，actの組の条件を満たすことで，$AttackState_{k+1} = AttackState_{END}$となる．つまり，任意の整数$n$に対し，攻撃者が最終目標に到達するということは，$AttackState_n = AttackState_{END}$を満たす必要がある．

C. $AttackPhase$および$Capability$

次に，式(3.11)の集合$AttackPhase$と集合Cp_kを具体化する．本手法では，ATT&CK®[93]を用いモデル設計した．ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge)は TTP (Tactics, Techniques and Procedures)を表現する知識ベースのオープンなモデルである．TTP は要素として攻撃者の戦術を意味する Tactics と，Tactics を達成するための技術を意味する Techniques を二軸とするマトリックスであり，攻撃者が共通的に利用する攻撃戦術と攻撃技術の知識を提供する．ATT&CK は多数の派生モデルが存在する．例えば，一般的な情報システム向けの TTP モデルである ATT&CK for Enterprise[94]や，制御システムに特化した TTP モデルである ATT&CK for ICS[95]がある．ATT&CK for ICS のマトリックスの例を Figure 3.3 に示す．

Figure 3.3　ATT&CK for ICS

この ATT&CK のモデルを基に，攻撃者の挙動モデルを具体的に表現する．攻撃者の戦術（ATT&CK Tactics）の要素とする有限集合を，$ATT\&CK_{Tac}$とし，攻撃者の技術（ATT&CK Techniques）の要素とする有限集合を$ATT\&CK_{Tech}$とする．この場合，$AttackPhase$，$Capability$は，それぞれ式(3.16)-(3.18)で表す．

$$AttackPhase = \{ p \mid p \in ATT\&CK_{Tac} \} \tag{3.16}$$

$$Capability = \{ c \mid c \in ATT\&CK_{Tech} \} \tag{3.17}$$

$$Capability_{p,v} = \{ c \mid c \in ATT\&CK_{Tech,p,v} \} \tag{3.18}$$

pをある攻撃フェーズとした場合，式(3.18)の$ATT\&CK_{Tech,p,v}$は，ターゲットvに対し実行できる ATT&CK Techniques から成る集合である．前述の通り，攻撃フェーズpに攻撃が成立するには，$ATT\&CK_{Tech,p,v} \neq \emptyset$を満たさなければならない．

D. 攻撃パスの生成

次に，$ThreatActor$が取る可能性がある攻撃パスについて述べる．$ThreatActor$は，攻撃開始時点の状態を$AttackState_0 = (p_0, v_0, Cp_0, lo_0, Cond_0)$とした場合，攻撃者の存在位置は$lo_0$にあり，そこから$ThreatActor$の最終ターゲット$v_{last}$に向かい，攻撃開始する．

lo_0の候補は，いくつか存在し，攻撃者の攻撃開始点とないうる物理エリアを要素とする集合，$PhysicalArea_{Dep}$，論理的な侵入減である外部のネットワークを要素とする集合$Network_{Ext}$，またはサプライチェーン攻撃による侵入の可能性があるコンポーネントを要素とする集合$Component_{Sup}$のいずれかである．これらの集合を用いると，lo_0は式(3.19)で示される．

$$lo_0 = \{e \mid e \in \{Component_{Sup}, PhysicalArea_{Dep}, Network_{Ext}\} \tag{3.19}$$

$TargetSystem$は，それを構成する各要素からなる無向グラフとして表現する．$TargetSystem$=$\{Component,\ PhysicalArea, Network\}$とした場合，グラフの頂点として$\{PhysicalArea, Component\}$の要素を用い，辺として$Network$の要素を用いる．このグラフを攻撃パスグラフと呼ぶ．

上記$TargetSystem$における攻撃パスグラフを$(PathGraph, Node\ V, Edge\ E)$とした場合，式(3.20)-(3.23)で表す．

$$PathGraph = \{V, E\} \tag{3.20}$$

$$V = \{Component, PhysicalArea, Dummy\} \tag{3.21}$$

$$E = \{Network, local\} = \{(Dummy, comp1), (area1, comp1), (comp1, comp2), (area2, comp2), \ldots\} \tag{3.22}$$

各有限集合が，式(3.23)-(3.25)により表現された$TargetSystem$のグラフ例を Figure 3.4 に示す．

$$Component = \{comp1, comp2, comp3, comp4, comp5\} \tag{3.23}$$

$$PhysicalArea = \{area1, area2, area3\} \tag{3.24}$$

$$Network = \{nw1, nw2, nw3, internet \mid nw1, nw2, nw3 \notin Network_{Ext}, internet \in Network_{Ext}\} \tag{3.25}$$

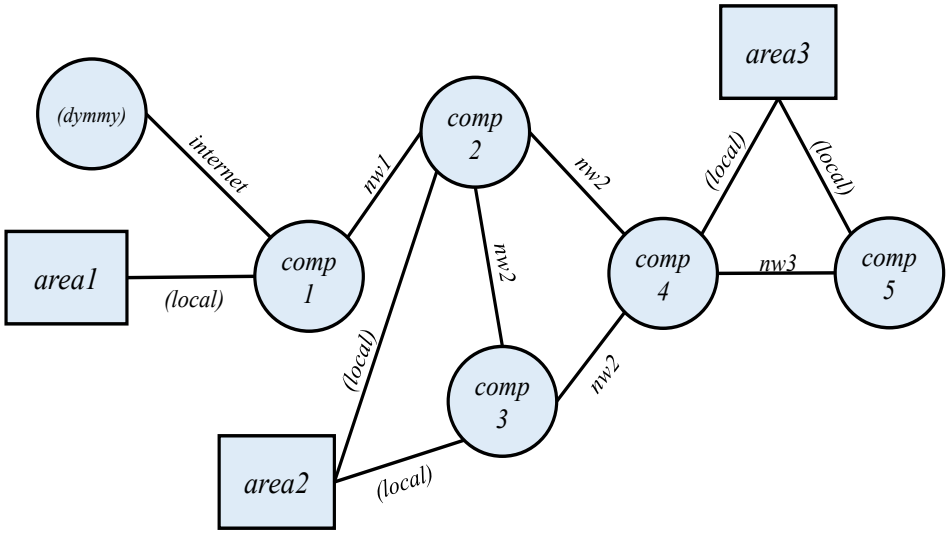


Figure 3.4 An example of $PathGraph$

$Component$は，単体で独立して存在しない場合を除き，$nw1, nw2, nw3, internet$の 4 つのネットワークのいずれかと論理的な接続関係がある．また，$internet$に代表される$Network_{Ext}$の要素との接続関係を定義する場合，その接続先となるコンポーネントは不明とみなし，代わりにダミーコンポーネント($dummy$)を定義する．ダミーコンポーネントは仮想的なコンポーネントノードとして機能する．また，各コンポーネントは，物理エリア$area1, area2, area3$のうち，いずれかのエリアに配置される．物理エリアとコンポーネントの関係は，ローカルアクセスを意味する．これは仮想的に存在するノード間の接続関係($local$)とみなす．

Figure 3.4 に示す$PathGraph$に対し，ある$ThreatActor$の攻撃パスの生成手順を以下に示す．ここで，$ThreatActor$は，初期位置$l_0 = area1$，最終目標は$(comp5, act)$と仮定する．

1. 最初のターゲット$v_0 = comp1$に対し，$l_0 = area1$から，攻撃を開始する．
2. $comp1$への攻撃が成功した場合，$comp1$を乗っ取り，そこから$comp2$に向かい攻撃試行する．攻撃ステップkにおいて，ターゲットv_k利用できる攻撃パターンが無い場合，すなわち，$AttackPhase_k$の要素Cp_kが式(3.15)を満たした際，これ以上の攻撃遂行は不可として攻撃パスの生成処理を終了する．
3. 攻撃目標とするコンポーネント$v_{last} = comp5$に到達し，最終目標$(comp5, act)$に該当する攻撃パター

ンactを遂行できた場合，攻撃は最終目標に到達したとみなし，攻撃パスの生成処理を終了する．

次に，初期位置l_0の該当する条件，および攻撃の成立条件について触れる．何らかの攻撃の成立条件は，攻撃する対象が持つ特徴（アトリビューション）により決まる．例えば ATT&CK Techniques では，攻撃技術の成立条件として，コンポーネントが持つ OS 等のソフトウェア（プラットフォーム）を規定している．例えば，ある攻撃技術に該当するプラットフォームが Linux の場合，Linux が動作するコンポーネントでなければ，その攻撃技術は成功しない．

次に攻撃条件のモデルを具体化する．例えば，あるコンポーネントが，自分自身を含む$TargetSystem$内の他コンポーネントのクレデンシャル情報，操作権限を保持している場合，攻撃の途中で攻撃者はそれらを取得し，攻撃条件を拡張して次の攻撃にそれを再利用する．第一にこのケースをモデル化する必要がある．

また，攻撃パスが成立するか否かを判断する観点において，特に物理的な侵入を前提とする場合，$TargetSystem$内の物理エリアに基づき，コンポーネントへのアクセス可否を決定する際にコンポーネントの物理エリアを明確化が必要である．

更に，コンポーネントは，ゲートウェイやネットワークスイッチなどのネットワークデバイスの含まれる．それらを攻撃対象とした場合，ネットワークデバイス固有の攻撃パターンとの対応付けが必要である．

以上より，$TargetSystem$要素の$Component$と，$Network$に対して，メタモデル（属性情報）を与える．各メタモデルを$Component_{Attr}$，および$Network_{Attr}$とした場合，その属性を組で表現したモデルを式(3.26)(3.27)に示す．

$$Component_{Attr} \coloneqq (name, pf, Admin, Cred, area) \tag{3.26}$$

$$Network_{Attr} \coloneqq (name, Ctldev, isExt) \tag{3.27}$$

式(3.26)が示す$name$は，コンポーネントの識別名を意味する．この識別名は，$TargetSystem$内において重複が無い．pfは，オペレーティングシステム（Windows や Linux 等）やミドルウェア等のコンポーネントにインストールされたソフトウェアを意味する．$Admin$ は，コンポーネントに対する操作権限から成る集合であり，少なくとも自身の捜査権限を有する．$Cred$は，コンポーネント内に存在するすべてのクレデンシャル情報である．$area \mid area \in PhysicalArea$は，コンポーネントの物理エリアを意味する．

式(3.27)の$name$は，$TargetSystem$内で重複の無いネットワークの固有名を意味する．$Ctldev$は，当該ネットワークを制御するデバイスである．$Ctldev$はシステムを構成するコンポーネントの一部であり，$Ctldev \in Component$を満たす．$isExt$は論理値であり，*True,* もしくは *False* のいずれかの値を取る．$isExt$が *True* の場合，$nw \in Network_{Ext}$であり，*False* の場合は，$nw \notin Network_{Ext}$を満たす．

E. 攻撃アクションの生成

$ThreatActor$は，$AttackState_k$の攻撃アクションが成功した場合，$AttackState_k$は，$AttackState_{k+1}$に状態遷移する． $AttackScenario$は，この$AttackState_k$の遷移履歴である．モデルの詳細を以下に示す．

I. $ThreatActor$の初期状態を$AttackState_0 = (p_0, v_0, cp_0, lo_0, cond_0)$とした場合，$p_0$は初期侵入(Initial Access)のフェーズであり，$TargetSystem$への初期の侵入を試みている状態を意味する．
II. 初期侵入後の状態$AttackState_k = (p_k, v_k, cp_k, lo_k, cond_k)$において，$v_k$が$v_{last}$でない場合，$ThreatActor$はできる限り内部侵入（横展開）を続ける．
III. 初期侵入後，あるいは横方向への展開を経て，攻撃者はターゲットv_k上でp_kを完了させるための攻撃を試行する．
IV. 攻撃コードを実行した後，$v_k \neq v_{last}$となる場合，次コンポーネントに向かい横展開する．逆に$v_k = v_{last}$となる場合，$Goal$で指定されるアクションを実行する．
V. $k \rightarrow k+1$の遷移で，横展開が成功した場合，攻撃者の存在位置は横展開する前のコンポーネント$(lo_{k+1} = v_k)$に設定される．そしてv_{k+1}は，次の攻撃対象コンポーネントとなる．
VI. $cond_k$は攻撃過程で攻撃者が入手したパラメータを要素とする集合である．$cond_k$は攻撃成功時に更新する．例として，コンポーネントcのクレデンシャル情報$cred_c$を攻撃者が入手した場合，$cond_{k+1} = \{cred_c\}$となる．$k+1$において，攻撃者は，攻撃試行時に搾取した$cred_c$を利用できる．更に，攻撃者がコンポーネントcの操作権限$admin_c$を入手した場合，$cond_{k+1} = \{admin_c\}$となる．攻撃者はこの条件の場合，$k+1$において，コンポーネントcの操作権限を攻撃時に利用できる．

攻撃フェーズは，$ATT\&CK_{Tac}$の要素間で定義された制約条件に基づいて遷移する．遷移ルールの例をFigure 3.5 に示す．この例の場合，攻撃者は$TargetSystem$に対してリモートまたはローカルから初期アクセス（Initial Access フェーズ）を試み，攻撃プログラムを実行（Execution フェーズ）する．ターゲットのコンポーネントが Goal で指定されている場合は，その目標を達成する攻撃（Impact フェーズ）を試行する．指定されたコンポーネントでなかった場合，当コンポーネントを対する情報収集（Discovery フェーズ），およびクレデンシャル情報の取得（Credential Access フェーズ）が行われる．次に，ネットワークの更なる探索（Discovery フェーズ）を行い，Goal で指定されたコンポーネントを見つけた場合，当該コンポーネントに対する攻撃（Impact フェーズ）を試みる．

既に該当するコンポーネントの操作権限がある場合は，その権限を使用して攻撃（Impact フェーズ）を行い，逆に無い場合はリモートで攻撃を試みる．リモートでの攻撃が不可能，または Goal で指定されたコンポーネントではない場合は，そのコンポーネントの操作権限を奪取する．これらのプロセスを Goal で指定されたコンポーネントに到達するまで繰り返す．

$ThreatActor$の特定の攻撃アクションモデルに適合するには，既存の ATT&CK モデルでは不十分であり，その拡張が必要とされる．ATT&CK が提供する戦術の中でも，特に Initial Access，Discovery，および Impact に関しては，ローカル経由とネットワーク経由の攻撃の違いが明確になっていない．この区別を明確にするため，これら戦術を「ローカル経由」と「ネットワーク経由」の分類を追加し，攻撃アクションモデルに組み込むことが求められる．この遷移ルールは$ThreatActor$ごとに異なり，$ThreatActor$に応じたフローチャートの再構成が必要である．例えば，Privilege Escalation を試行する$ThreatActor$は，Privilege Escalation を含むフローチャートになる．

以上のプロセスに従い，定義された攻撃シナリオに沿って$TargetSystem$上の攻撃者のアクションを再現する．

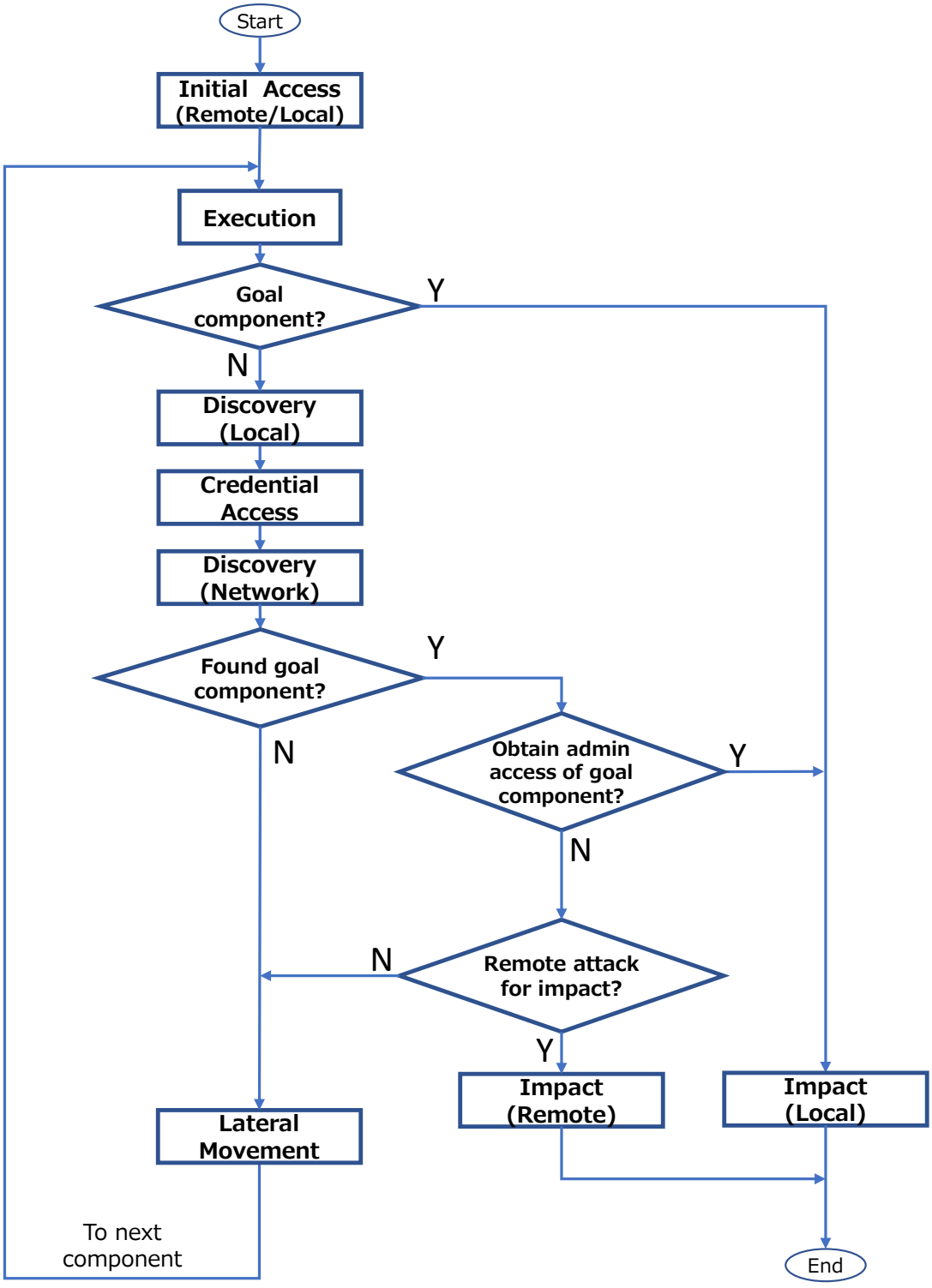


Figure 3.5 Example of a transition rule of *ThreatActor*

3.3. 評価

3.3.1 評価対象モデルの概要

本提案手法の評価のための制御システムモデルを Figure 3.6 に示す．前節で述べた攻撃モデルから攻撃シナリオを生成する．評価対象モデルのコンポーネントの各インスタンスを Table 3.1 に示す．これらコンポーネントは，式(3.26)に示すメタモデルを含み，の各列は，各コンポーネントのメタモデルである．列 Name は，コンポーネントの識別名であり，列 Platform は，コンポーネントのプラットフォームソフトウェア，列 Admin は，コンポーネントの操作権限，列 Credential は，認証クレデンシャル情報，そして列 Area は，コンポーネントの物理的な位置を示している．

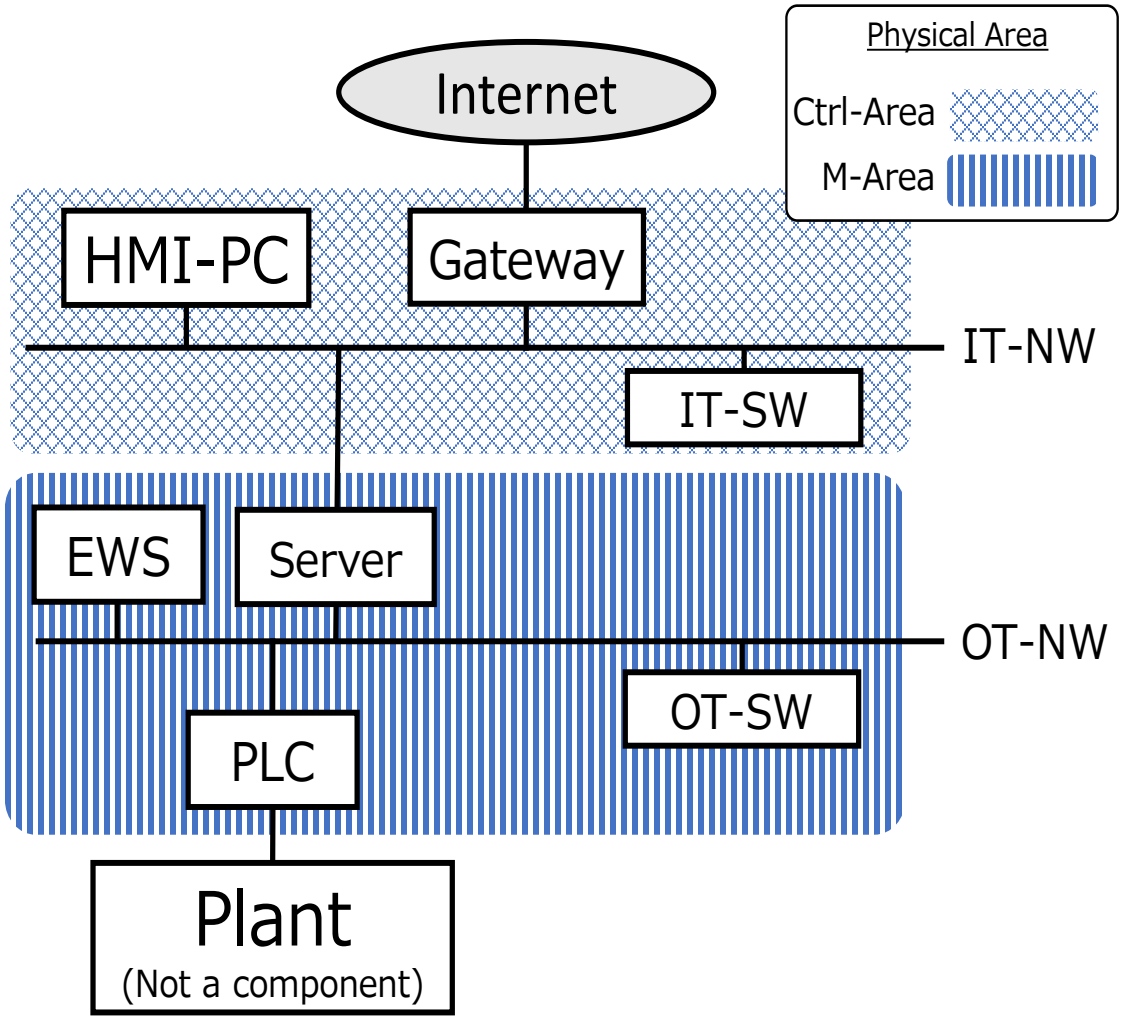


Figure 3.6　Image of the system to be evaluated

Table 3.1 List of $Component$

#	Name	Platform	Admin	Credential	Area
1	HMI-PC	Windows	$Admin_{HMI\text{-}PC}$	$cred_1$	Ctrl-Area
2	Server	Linux	$Admin_{Server}$	$cred_1$	M-Area
3	EWS	Windows	$Admin_{EWS}$ $Admin_{PLC}$	$cred_1$	M-Area
4	PLC	Other	$Admin_{PLC}$	-	M-Area
5	Gateway	Linux	$Admin_{Gateway}$	-	Ctrl-Area
6	IT-SW	Other	$Admin_{IT\text{-}SW}$	-	Ctrl-Area
7	OT-SW	Other	$Admin_{OT\text{-}SW}$	-	M-Area

Table 3.2 は，評価対象システムのネットワーク一覧を示す．各ネットワークは式(3.27)に示すメタモデルを示し，各列は，メタモデルを示す．列 Name は，各ネットワークの識別名，列 ControlDevice は，ネットワークの制御デバイス，列 External?は，外部のネットワークであるか否かを判定する論理値である．

Table 3.2 List of $Network$

#	Name	ControlDevice	External?
1	Internet	-	True
2	IT-NW	IT-SW	False
3	OT-NW	OT-SW	False

Table 3.3 は，$ThreatActor$の$Adversary$，および$Goal$を示す．この例の場合，列 Adversary に示す APTxx という名称の攻撃者であり，Loss of Control（制御の喪失）を最終目的とした．$Goal$は式(3.8)，および Table 3.3 に従い，Target と Technique の組として表現される．例えば，$Goal$ = (PLC, "T1485")で表される場合，PLC に対し，データ破壊(T1485- Data Destruction)の遂行が最終目標となる．

Table 3.3 $Adversary$ and $Goal$ in this evaluation

Adversary	Goal	Target	Techniques
APTxx	Loss of Control	PLC	TA0040 - Impact (Local/Remote) - T1485 - Data Destruction - T1565 - Data Manipulation - T1499 - Endpoint Denial of Service

Table 3.4 は，$ThreatActor$が有する$Capability$である．これは式(3.18)に準じ，ATT&CK Techniques と，各 Technique に対応する ATT&CK Tactics から構成される．この$ThreatActor$は, Dragonfly[96], Lazarus[97], Darkside[98]などの ICS に対する複数の攻撃事例に基づく仮想的な APT グループとする．

Table 3.4 A list of $Capability$ in this evaluation

#	Tactics	Techniques
1	TA0001 - Initial Access (Remote)	T1189 - Drive-by Compromise T1133 - External Remote Services T1566 - Phishing
2	TA0001 - Initial Access (Local)	T1199 - Trusted Relationship T1078 - Valid Accounts
3	TA0002 - Execution	T1059 - Command and Scripting Interpreter T1203 - Exploitation for Client Execution T1204 - User Execution
4	TA0007 - Discovery (Local)	T1087 - Account Discovery T1201 - Password Policy Discovery T1518 - Software Discovery
5	TA0006 - Credential Access	T1110 - Brute Force T1555 - Credentials from Password Stores
6	TA0007 - Discovery (Network)	T1040 - Network Sniffing T1049 - System Network Connections Discovery
7	TA0008 - Lateral Movement	T1210 - Exploitation of Remote Services T1021 - Remote Services T1550 - Use Alternate Authentication Material
8	TA0040 - Impact (Local)	T1485 - Data Destruction T1565 - Data Manipulation T1499 - Endpoint Denial of Service
9	TA0040 - Impact (Remote)	T1565 - Data Manipulation T1499 - Endpoint Denial of Service

3.3.2 処理手順および攻撃シナリオの生成結果

前節に示された対象システムを基に，シナリオ生成処理手順を示す．

1. $TargetSystem$を定義する．
2. $TargetSystem$をグラフ化する．
3. $Goal$,$Capability$から，$ThreatActor$モデルを定義する．
4. $TargetSystem$，および$ThreatActor$に基づき，$AttackState_0$を定義する．攻撃者の$Capability$に基づき，攻撃起点となる$TargetSystem$内の要素のいずれかを初期位置l_0として指定する．

5. Figure 3.5 に示すフローチャートに基づき，$AttackState_0$を開始状態とし，$Goal$が達成する，もしくは攻撃できなくなるまでアクションを実行する．攻撃成立の判定条件は，$AttackState_k$において式(3.15)を評価することである. 式 (3.15)を成り立つ場合, 攻撃フェーズが次に遷移する. 逆に, 式 (3.15)を満足しない場合は，以降の攻撃は不可とみなし，シナリオ生成を終了する．

今回の評価で生成した攻撃シナリオの一例を Figure 3.7 に示す.

AttackScenario I

AttackState[0]: {point: Internet, target: HMI-PC, phase: INITIAL_ACCESS(Remote), conds: {creds: [], admins: []}}
AttackState[1]: {point: HMI-PC, target: HMI-PC, phase: EXECUTION, conds: {creds:[], admins: []}}
AttackState[2]: {point: HMI-PC, target: HMI-PC, phase: DISCOVERY(Local), conds: {creds: [], admins: [HMI-PC]}}
AttackState[3]: {point: HMI-PC, target: HMI-PC, phase: CREDENTIAL_ACCESS, conds: {creds: [], admins: [HMI-PC]}}
AttackState[4]: {point: HMI-PC, target: ???, phase: DISCOVERY(Network), conds: {creds: [cred1], admins: [HMI-PC]}}
AttackState[5]: {point: HMI-PC, target: Server, phase: LATERAL_MOVEMENT, conds: {creds: [cred1], admins: [HMI-PC]}}
AttackState[6]: {point: Server, target: Server, phase: EXECUTION, conds: {creds: [cred1], admins: [HMI-PC]}}
AttackState[7]: {point: Server, target: Server, phase: DISCOVERY(Local), conds: {creds: [cred1], admins: [HMI-PC, Server]}}
AttackState[8]: {point: Server, target: Server, phase: CREDENTIAL_ACCESS, conds: {creds: [], admins: [HMI-PC, Server]}}
AttackState[9]: {point: Server, target: ???, phase: DISCOVERY(Network), conds: {creds: [cred1], admins: [HMI-PC, Server]}}
AttackState[10]: {point: PLC, target: PLC, phase: IMPACT(Remote), conds: {creds: [cred1], admins: [HMI-PC, EWS, Server, PLC]}}

AttackScenario II

AttackState[0]: {point: M-Area, target: HMI-PC, phase: INITIAL_ACCESS(Local), conds: {creds: [], admins: []}}
AttackState[1]: {point: OT-SW, target: HMI-PC, phase: EXECUTION, conds: {creds: [], admins: []}}
AttackState[2]: {point: OT-SW, target: HMI-PC, phase: DISCOVERY(Local), conds: {creds: [], admins: [OT-SW]}}
AttackState[3]: {point: OT-SW, target: HMI-PC, phase: CREDENTIAL_ACCESS, conds: {[creds: [], admins: [OT-SW]}}
AttackState[4]: {point: OT-SW, target: ???, phase: DISCOVERY(Network), conds: {creds: [], admins: [OT-SW]}}
AttackState[5]: {point: EWS, target: EWS, phase: LATERAL_MOVEMENT, conds: {creds: [], admins: [OT-SW]}}
AttackState[6]: {point: EWS, target: EWS, phase: EXECUTION, conds: {creds: [], admins: [OT-SW, EWS]}}
AttackState[7]: {point: EWS, target: EWS, phase: DISCOVERY(Local), conds: {creds: [], admins: [OT-SW, EWS]}}
AttackState[8]: {point: EWS, target: EWS, phase: CREDENTIAL_ACCESS, conds: {creds: [], admins: [OT-SW, EWS]}}
AttackState[9]: {point: EWS, target: ???, phase: DISCOVERY(Network), conds: {creds: [cred1], admins: [OT-SW, EWS]}}
AttackState[10]: {point: PLC, target: PLC, phase: IMPACT(Local), conds: {creds: [cred1], admins: [OT-SW, EWS]}}

Figure 3.7　Attack scenario examples obtained in this study

最初のシナリオ(AttackScenario I)は，外部のネットワークである Internet を起点とする攻撃シナリオである．攻撃者は，Internet から HMI-PC，IT-NW，Server，OT-NW，PLC という流れで攻撃対象を切り替えつつ$Goal$に向け，攻撃を実行する．次のシナリオ(AttackScenario II)の場合，物理エリアである M-Area を起点とするシナリオである．このシナリオでは，M-Area から始まり，次に OT-NW Switch，OT-NW，EWS，最後に PLC という順で進行する．

AttackScenario II の場合，PLC への攻撃は，$Admin_{PLC}$（EWS から入手した操作権限）を利用する．事前情報を有さない攻撃者にとって，DISCOVERY(Network) フェーズに到達した時点の次ターゲットは不定である．これは，DISCOVERY(Network) フェーズの終了時点で確定し，ネットワークに接続するコンポーネントのうち，いずれかとなる．今回の例では EWS となる．攻撃者は，各フェーズにおいて$Capability$

に指定されたパターン，すなわち Table 6 の Techniques を基に，攻撃状態の一連から成るプロセスを生成する．Figure 3.8 に示すいずれの攻撃パターンが利用できる場合，当攻撃フェーズは達成可能とし，次フェーズに状態遷移する．

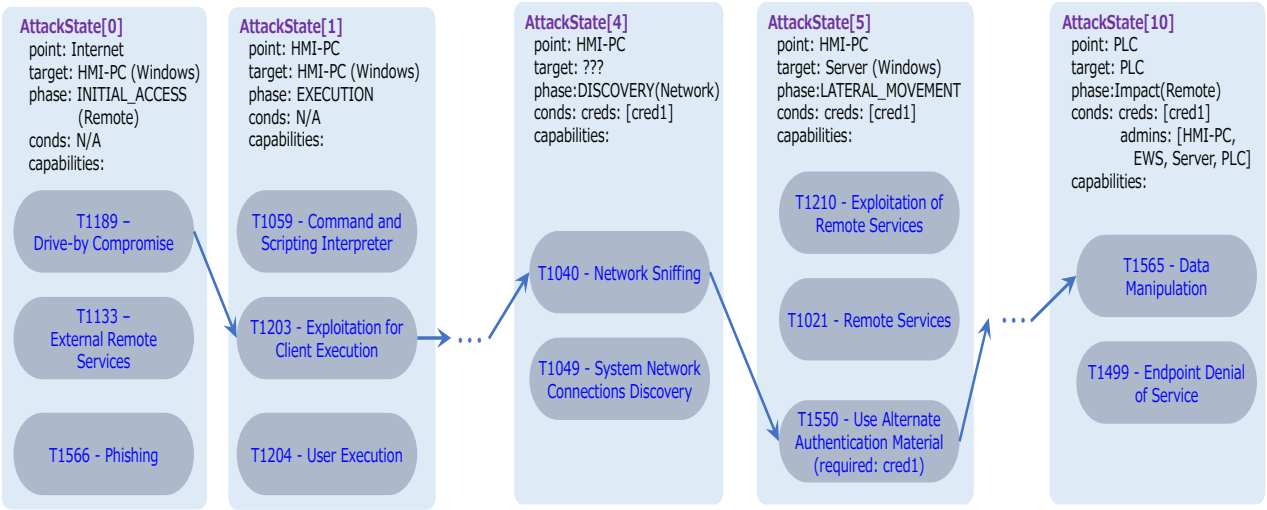


Figure 3.8　Graphical image of attack techniques selection on each phase

攻撃シナリオの中には，攻撃過程で攻撃の遂行に必要とする追加条件が求められる場合がある．例えば，入手済みの情報を再度利用する Use Alternate Authentication Material (T1550)を始めとする攻撃技術が例である．このようなケースの場合，該当する攻撃フェーズにおいて，対応する攻撃条件を満足する場合に攻撃が達成可能とする．

以上より，提案するモデルにより，$ThreatActor$による攻撃がシステム内で進行するメカニズムを構造的に表現でき，$ThreatActor$のモデルに応じて$TargetSystem$への攻撃を再現できることを確認した．

3.3.3　計算量の検証

次に，システム規模と，計算量について議論する．$TargetSystem$，$ThreatActor$の構成要素の数が増加するにつれ，攻撃フェーズ毎に取り得る組み合わせ数も増加する．本提案モデルに基づく攻撃シナリオの得るために必要な計算量は，得られる攻撃シナリオの攻撃フェーズ平均数をN_{Ph}とし，$ThreatActor$が，各攻撃フェーズに，実行できる攻撃パターンの平均数をN_T，攻撃対象の平均個数をN_Cと置くと，単純な組み合わせ計算の場合，その計算量は$O({N_T}^{N_{Ph}} \times {N_C}^{N_{Ph}})$となる．すなわち，$N_{Ph}$に対して計算量は指数的に増加する．$N_{Ph}$は，システムの規模が大きくなるにつれ増加することため，結果として組み合わせ爆発が起こり，現実的な時間内に計算処理が完了できない可能性がある．この評価のため，以下に置く仮定で計算時間を評価した．

I.　評価対象システムは Figure 3.6 とする
II.　評価対象システムにおいて可能性のある攻撃シナリオを全て列挙する
III.　評価を簡略化するため，システム内のネットワークは Table 3.2 の 3 つのインスタンスのみとし，各

ネットワークに接続するコンポーネントのみが増加する

IV. コンポーネント数の加算単位は，Table 3.1 記載のコンポーネントからなるセットであり，システム規模はコンポーネントの自然数（N 倍）分乗じた単位で増加する

V. 攻撃シナリオの生成過程では，Table 3.3，Table 3.4 で定義された脅威アクターが，ネットワーク上の各コンポーネントに対して総当たり的に攻撃を試みる．攻撃ステップは，脅威アクターの最終目標が達成されるか，敵に利用可能なさらなる攻撃方法がなくなるまで続ける

以上の仮定で, 攻撃シナリオの特定とそれらのリスト生成に必要な計算処理時間を評価した. 計算環境を Table 3.5 に示す.

Table 3.5　Computing environment in this study

Item	Detail
CPU	Intel Corei7-8665U 2.11GHz (4 cores/8 threads)
Memory	DDR4 32GB memory
OS	Windows 10 Pro (Ver 10.0.19042.1237)
Programming Language	Python3.9
Graph Algorithm	NetworkX 3.1

コンポーネントの数が異なる標的システムに対する計算処理時間を Figure 3.9 に示す．X 軸は標的システム内のコンポーネント数を示し，Y 軸は上記プロセスの計測された計算時間を示す．各プロットは 10 回の実験から得られた平均計算時間である．グラフ内の点線は，プロットから計算された指数近似曲線 ($R^2 \sim 0.9767$)である．

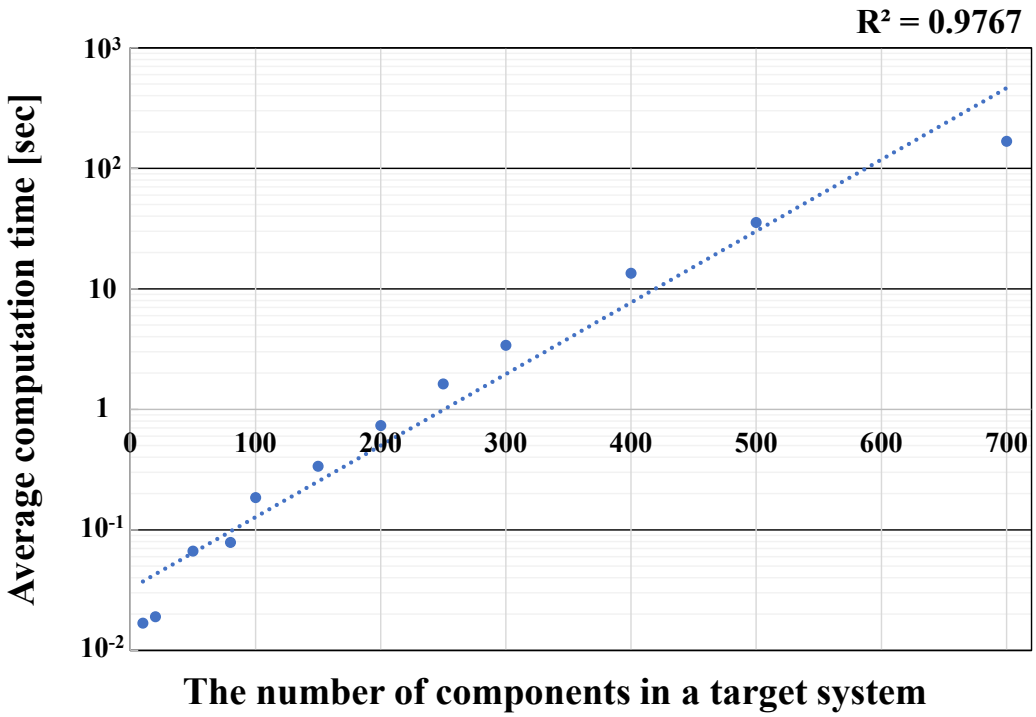


Figure 3.9　Graph of the average computation time and the number of components

この結果から，Table 3.5 に示した計算環境の場合，標的システム内のコンポーネント数に応じて，計算時間は概ね指数オーダーで増加する傾向となった．これは，$TargetSystem$のコンポーネント数が増加するほど計算量が指数的に増加し，大規模システムの場合，組み合わせ爆発により現実時間で完了できない計算量になる可能性がある．計算時間は，コンポーネント数だけでなく，ネットワークのグラフ構造，$ThreatActor$の属性，およびコンポーネントが持つ属性にも依存し，要素数が増えるにつれて計算量は指数オーダーで増加する．

3.4.　　考察

提案する表現モデルにより，脅威事象を，制御システム内の攻撃者の挙動として構造的に表現できる．この結果から，脅威事象に関する知識の再利用性と，脅威シナリオの優先順位付けにおける有用性を考察する．まず，脅威事象に関する知識の再利用性の考察を述べる．$ThreatActor$を本手法を基にパターン化することで，本手法のモデルに基づき表現された$TargetSystem$に対し，パターン化された$ThreatActor$の挙動を再現できる．これは，システム構成が異なる場合であっても，同じ$ThreatActor$の挙動を再現できることを意味する．攻撃フェーズは遷移する条件に関しても同様，式(3.12)の$AttackState_k$のモデルに応じて，各攻撃の成功条件を$Capability$として事前定義し，それをテンプレート化することで再利用性を確保できる．

$ThreatActor$は実際に存在する攻撃グループ等の事例を踏まえて，それぞれの条件を定めてパターンを定義する．$ThreatActor$は，設定可能なパラメータの自由度が大きい分，パターン定義の作業の際に，却って属人性の増加が予想される．この課題に対し，ATT&CK を始めとする標準的に定義された知識モデ

ルを元に定義することが一案である． ATT&CK を例にした場合，"ATT&CK Groups"と呼ばれる標準的な TTP のモデルが公開されている．例えば， Dragonfly[96]などの制御システムに特化した攻撃者グループの情報が利用できる．これを$ThreatActor$の属性定義時に利用することにより，$ThreatActor$のパターン定義において属人性を排除できる．欠点としては，インスタンス化されていない未知の$ThreatActor$に対しては，$ThreatActor$のモデルを事前に構築しない限り，脅威シナリオを得ることができない．実際，$ThreatActor$は，日々新しいものが多く出現しており，未知の$ThreatActor$への対処の本提案手法への組み込みは課題の一つである．

また，脅威シナリオの優先順位を決定する観点における有用性を考察する．本提案モデルは，$TargetSystem$，および$ThreatActor$の各モデルを構築し，これらモデルから$TargetSystem$のグラフを基に網羅的に攻撃パスを算出し，その中のうち$ThreatActor$が最終目標$Goal$に到達する攻撃シナリオの存在有無を評価する．網羅された攻撃パスの中で，到達しうるシナリオが存在する場合，$ThreatActor$によって$TargetSystem$への攻撃が成功する可能性があることを立証できる．網羅されたシナリオの中で「最終目標$Goal$に到達するか否か」という観点で，そのシナリオが実現する可能性(Likelihood)を評価できるため，関係者間で優先すべきシナリオの合意が得られやすい．Likelihood の定量化に関しては，次章にて詳しく述べる．

本提案手法の脅威識別の限界について，考察する．本提案手法でモデル化される脅威事象は，既存知識に基づき表現される．つまり，ゼロデイ攻撃のような，過去に報告が無く，既存知識が存在しないタイプの脅威に対するモデル化は制約がある．ゼロデイ攻撃は大きく分けると二種類あり，一つはゼロデイ攻撃の実体が，本提案手法においてモデル化された抽象的な攻撃パターンのモデルとして分類可能なものと，そうでないものが存在する．前者の場合，本提案手法において表現可能であることから，前者の種別に該当するゼロデイ攻撃の場合は，表現することが可能となる．ただし，その場合，当ゼロデイ攻撃を利用する$ThreatActor$を定義，もしくは再定義する必要がある．

後者の場合，すなわち，既存の攻撃手法とは全く異なる概念での攻撃パターンの場合，本提案手法で表現することは限界がある．これを解決するには，ゼロデイ攻撃の動向を観測し，本提案手法の前提となる$AttackPhase$，および$Capability$を更新し，当該ゼロデイ攻撃を実現する$ThreatActor$を定義することで，対処できる可能性がある．そのためには，脅威インテリジェンスサービスシステムとの連携が必要であるが，脅威インテリジェンスサービスから提供される STIX[99]で表現された脅威情報を基に，リアルタイムに$ThreatActor$を定義することで実現できる可能性がある．

更にモデル化誤差も残る課題である．本提案手法では，TTP の標準モデルであり ATT&CK を採用している．TTP はサイバー脅威の過去事例に基づき，攻撃者の活動（戦略，技術，手順）の抽象モデルである．実際の攻撃者の活動は，具体的な攻撃ツール・スクリプト，実存するソフトウェアの脆弱性を応用するため，モデルの精度を高めるためには，TTP より詳細のモデルでモデル化する必要がある．Figure 3.10 に示す Pyramid of Pain は，サイバー脅威検知の困難さを表現したモデルである．このモデルは，脅威対象が攻撃過程で発生させる証跡が具体的であるほど検知は容易であるが，攻撃ツール，TTP と脅威対象の攻撃モデルが抽象的になるほど検知が困難になることを表現したものである．一つの TTP で表現される攻撃シナリオであっても，その具体的なインスタンスは複数想定され，中には，実際に攻撃に応用できないインスタンスもあり得る．これを解決するには，サイバー脅威情報の収集と，サイバー攻撃の実体と TTP といった抽象的な攻撃モデルとの関連付けられた知識モデルを構築することが一案であるが，それ

を効率的に実現する手法の確立が課題である．

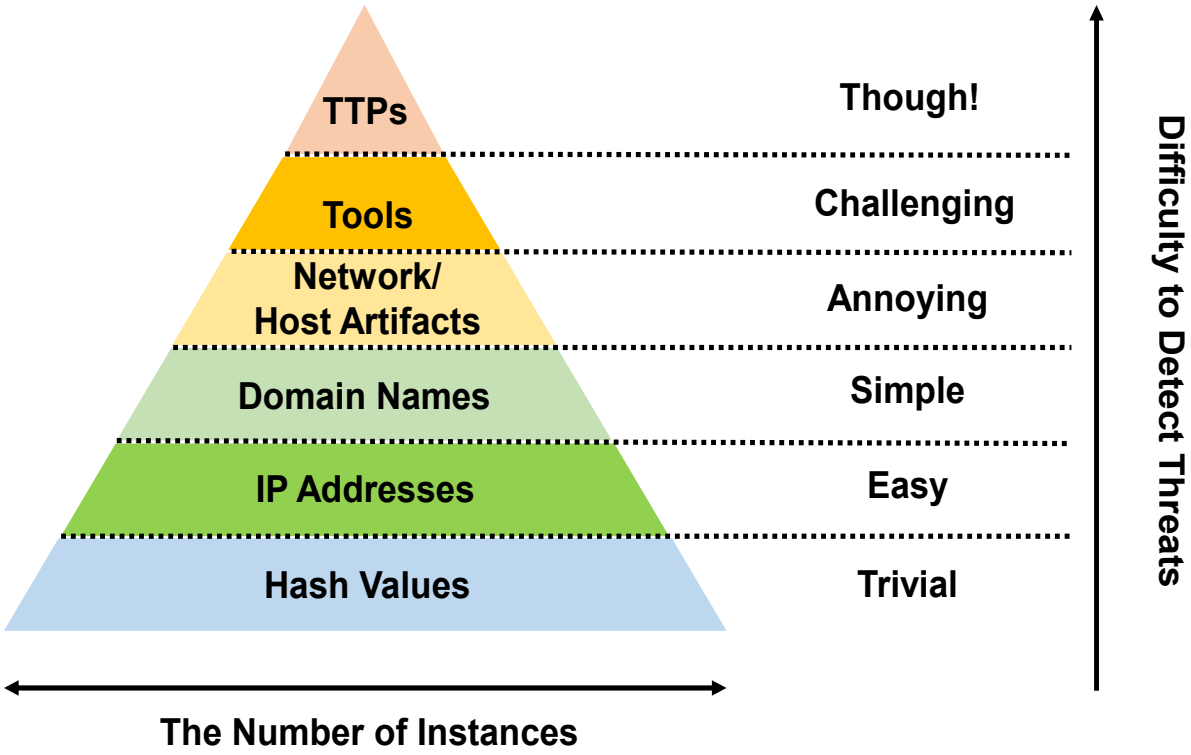


Figure 3.10　Pyramid of Pain

本提案手法により，属人性の排除，または自動化を達成した項目について Table 3.6 に示す．最初に，脅威アクターの識別は，本提案手法により達成できる．本手法により，脅威アクター再利用性を確保することができる．また，想定される攻撃シナリオの識別については，グラフを網羅的に探索することで，自動的に識別することができる．一方で，攻撃が成功した結果としての影響，すなわち，脅威アクターの最終目標を，属人性を排除し，自動的に識別することは達成できない．これは，個々の制御システムの機能，環境，運用形態といった，個々の状況に依存するためである．脅威アクターの最終目標識別の自動化は，今後検討の余地がある．

Table 3.6　Achievement status of elimination of individuality or automation by the proposed method

#	Process	Description	Achievement
1	**Identify threat actors**	Identify assumed threat actors	Completed
2	**Identify impactful events**	Identify impactful events for the system owner of the target system	None
3	**Identify of attack scenarios**	Obtain a comprehensive set of attack scenarios for the target system	Completed

最後にモデルの要素数と計算時間について考察を述べる．先行研究では，ネットワークのノード数nに関連して攻撃グラフを生成する計算時間の調査報告がある[101]．この報告によると，アルゴリズムによって異なるが，攻撃グラフの生成に必要な計算量は，$O(2^n)$から，少なくとも$O(nlog(n))$までの範囲である[102][103][104]．これらは提案されたアプローチとは異なるモデルに基づくが，本提案方式もグラフ探索を通じて攻撃シナリオを導出することから，同様の計算量の傾向が得られると予想される．本研究の実装では，計算時間の増加傾向が Figure 3.9 で観察されるように，計算量は$O(2^n)$と$O(\mathrm{n}^m)$の間にある($m \in \mathbb{N}$)と予想される．

計算量を削減する方法として考えられるのは，攻撃者の中間状態と，脅威インテリジェンスなどの過去事例の知識を基に，攻撃フェーズ毎で攻撃者が利用する攻撃パターンに対して重みを定義する方法[105]が考えられる．その例として，$ThreatActor$の持つパラメータに対して，意思決定モデルを定義する．その定義にあたり，脅威インテリジェンスに基づく過去事例の統計情報や，ゲーム理論[106]に基づく意思決定アルゴリズムを導入する．この意思決定モデルを基に，$ThreatActor$がとる攻撃状態に対し，攻撃パターンの重みを意思決定アルゴリズムにしたがって決定する．これにより，優先対処すべきシナリオの機械的な抽出が実現できる可能性がある．また，計算量の削減手法として，攻撃グラフの生成時にノードを取捨選択することで実現する報告者らの研究チームで研究報告があり，少ない計算量で優先的な攻撃シナリオを得られることを報告した[107]．本提案方式に基づく更なる計算量削減を実現する方法として，前述の脅威インテリジェンスやゲーム理論などの活用も含め，引き続き検討する余地がある．

3.5. 本章のまとめ

本研究では，脅威モデリングの脅威識別過程において，脅威モデルの再利用性を確保と，属人性を排除しつつ真に優先的に対処すべき脅威シナリオを実現する脅威シナリオモデル化手法として，リスク識別オントロジに基づく制御システムに対して攻撃挙動を表現する新しい構造モデルを提案した．提案モデルは，先ず Diamond model を基に，制御システムを物理サイト，ネットワーク，およびコンポーネントを要素とする集合論的な構造モデルとして表現する．次に，脅威アクターの構造モデルとして表現された制御システム上の動的な挙動を状態遷移モデルとして定義する．各モデルの制約条件から，脅威事象やシステムの特性に応じた攻撃成立条件を，共通化された知識モデル，過去事例を基にオントロジを構築し，モデル化することで，脅威モデルの再利用性を確保する．これにより，多様な構成の制御システムに対して脅威アクターの挙動に基づく攻撃シナリオと，最終目標とする脅威アクターが目標とする制御装置などの制御システム資産への到達性を決定する．この提案モデルを脅威モデリングにおけるリスク識別過程に適用することで，そのプロセスの属人性を排除しつつ，脅威シナリオが成立する過程を構造的に表現でき，シナリオの妥当性判断が容易となる．また，制御システムの構成を，本提案モデル定義の制約の元，組み替えることで，パターン化された脅威アクターの挙動を別構成の制御システムであっても再現でき，脅威シナリオの成立可否を機械的に判別可能となる．提案モデルの課題として，一つは組み合わせ爆発がある．これは，攻撃者が各攻撃状態で取り得る行動の判断モデルを導入することで改善できる可能性がある．更に脅威モデルの誤差の解決についても今後の課題である．

優先的な脅威シナリオを獲得する観点では，本手法では，$ThreatActor$が最終目標への到達可否を評価することで，現実化する可能性がある脅威シナリオを得ることができる．一方で，最終目標への到達可能

な複数のシナリオの優先順位付けの課題が残る．それを解決するには，$ThreatActor$による攻撃シナリオの成功確率を定量的に表現する必要である．定量化に関する議論は次章にて詳説する．

第4章 リスク評価：オントロジ駆動型モデリングに基づくリスク定量化

4.1. はじめに

前節では，制御システムに想定される膨大な脅威シナリオの内，モデル化された脅威アクターによって最終目標に達成する恐れがある脅威シナリオの抽出できる手法を示した．一方で，抽出された脅威シナリオの中でも優先順位が存在する．そのため，脅威モデリングにおいて，次に実施することは，各脅威シナリオの優先順位を決定することである．脅威シナリオの優先順位は，各シナリオのリスクの大小により決定される．前章において，脅威アクターの最終目標への到達可能性を以て，脅威シナリオの優先順位を決定できる可能性を示したが，更に厳密にリスク評価するには，リスクを定量表現する必要がある．

脅威シナリオの優先順位は，脅威シナリオのリスクで決まる．リスクとは，脅威事象が現実に発生した時の影響度合い(Severity)と，脅威事象の発生しやすさ度合(Likelihood)により決まる[71]．特に Likelihood は，実例に基づく客観的な評価を求められるが，標準で定義されている基準の多くは曖昧である．その結果，Likelihood の評価が属人化してしまう[36]．例として IEC 62443-2-1:2010[129]では，Likelihood の基準として，Table 4.11 に示す定義がある．この指標従い，脅威事象の Likelihood を決定すると，その基準の曖昧性が原因で評価者による解釈が発生する．その結果，Likelihood が属人化する．

Table 4.1 Example of likelihood in IEC 62443-2-1:2010

Likelihood	Description
High	A threat/vulnerability whose occurrence is likely in next year
Medium	A threat/vulnerability whose occurrence is likely in next 10 year
Low	A threat/vulnerability for which there is no history of occurrence and for which the likelihood of occurrence is deemed unlikely

この課題に対し，基準の曖昧性を排除するために複数の標準的な基準を組み合わせて評価指標を具体化するアプローチが報告されている．先行研究の例として，CVSS (Common Vulnerability Scoring System)[109]と，CWSS (Common Weakness Scoring System)[110]を組み合わせた Likelihood の評価手法が報告されている．CVSS は脆弱性の深刻度を計算するシステムであり，NVD[111]を始め，多くの脆弱性情報配信システムに採用されている．CWSS は，脆弱性の特性や攻撃界面のなりやすさ，脆弱性が置かれた環境的制約に応じて脆弱性の深刻度を 0 から 100 までの整数スコアで表現する．

CVSS や CWSS に基づく多くの先行研究の提案法は，脆弱性のリスクを計算するシステムである CVSS，CWSS の基準から Likelihood を決定するものであり，IEC 62443-2-1:2010 などの定性的な指標と比べ，その判断基準を明確化する手法である．例として，CVSS の基準を脅威シナリオのリスク値算出に適用する手法[57][58]や，CVSS と CWSS を組み合わせてリスクの定量化に関する研究が報告されている[59]．

CVSS の指標について簡単に説明する. Table 4.22 に, CVSS の基準の一つである AC（Attack Complexity：攻撃複雑度）を示す.

Table 4.2　The definition of AC in CVSS

AC	Description
High	• It is necessary to collect information in a suspicious manner such as privilege escalation or obfuscation before attacking • Attack is possible only when the target system has specific settings
Medium	• Some information needs to be collected before the attack • Specific settings with non-standard are required to exploit
Low	• Systems can be attacked without any special conditions

これらの基準を判定するためには，脅威事象を Table 4.22 に示される基準のいずれに該当するか明確になるまで具体化する必要がある．この基準を利用し，関係者間で Likelihood 評価結果の合意が得るためには，判断の根拠が自明なるように脅威事象を詳細に表現しなければならない．言い換えると，脅威事象が抽象的な表現であるほど，Likelihood の判断が難しくなる．その結果，リスク値の評価結果が属人化してしまう.

上記を踏まると，脅威シナリオのリスクの定量的な評価は困難であり，評価者の主観，解釈が必ず含まれてしまう．したがって，属人性排除は現状困難である．一方で，脅威シナリオを具体的に表現した場合は，要素毎に Likelihood をでき，詳細なリスク評価とリスクの定量化が達成できる可能性がある.

リスクの定量的な表現の既存手法として，攻撃シナリオの要素に確率論的な定量的指標を導入し，マルコフモデル[112]やベイジアンネットワーク[113]のような構造モデルで表現する取り組みも報告されている．これらの手法は各要素の確率モデルの決定するプロセスについては触れられていない．また，X. Ou らよって提案された手法では，攻撃グラフで表現されたシナリオに対して，各脆弱性の CVSS を元にリスク値を決定する手法を提案されているが，この手法は脆弱性情報のみを扱っており，攻撃者自身の要素を考慮していない[114].

前章に示した，オントロジ駆動型モデリングに基づく脅威事象の表現化手法は，攻撃者の挙動をモデル化し，脅威シナリオの内部構造を明示的に表現できる．すなわち，このモデルを利用し，オントロジに基づき表現された攻撃者の挙動を数学的に表現することで，リスクモデルの再利用性を確保しつつ，定量的な尺度で表現できる可能性がある．本章では，オントロジ駆動型モデリングに基づき，前章に示した脅威シナリオ表現手法を応用したリスクの定量化モデルについて議論する.

4.2.　リスクの定量化モデル

リスク管理フレームワークである ISO 31000:2018 の定義に従うと，リスクは「目標に対する不確実性の影響」であり，その影響の程度をリスク値として表現する．セキュリティの文脈では，特定の脅威シナ

リオが脅威に対して脆弱なエンティティに与える影響の程度と解釈できる．これは，脅威シナリオのリスク値が高いほど，それに関連するリスクも大きいという自明な事実を表す．リスク値は，式(4.1)に示される通り，影響と可能性の積として表現される．リスクのモデルは，多様性があり，式(4.1)に示す他，様々なモデルが報告されている[115][116]が，本論では，最も合意形成が取れている ISO 31000:2018 の定義に従う．

$$Risk = Impact \times Likelihood \tag{4.1}$$

$Impact$は，攻撃の結果として制御システムオーナが直面する負の結果を指す．例えば，制御システムの停止による業務中断，安全性と環境への影響，および営業秘密の抽出に関連する事業機会の損失などが挙げられる．$Likelihood$は，前述の負の結果が実現する確実性を意味する．セキュリティの文脈では，攻撃の成功の確率を意味する．リスクの大きさは，脅威イベントが実現した場合の影響の重大さとその発生の可能性の両方に比例する．その結果，関連するリスクが高い脅威イベントが優先される傾向となる．提案された方法でリスク値を計算することにより，脅威イベントの構造がより具体的になり，リスク値の定量的表現が可能になることが期待できる．本節では，前章に示したモデルにより表現された脅威シナリオのリスクについての定量的表現を試みる．

前章に示したモデルにより表現された脅威シナリオによると，$ThreatActor$は，複数の$AttackState$を経て一連の攻撃活動を遂行し，特定エンティティを標に侵入を試みる．$ThreatActor$は，最終目標に到達すると，攻撃が実行される．標的とする攻撃者の目的は場合によるが，制御システムの文脈では，$ThreatActor$の主な目的は，制御システムの稼働停止や運用の妨害，または生産および制御プロセスに関する営業秘密の搾取などがある．一連の攻撃シナリオを考えると，攻撃中に被害が発生する可能性も少なからず存在するが，攻撃シナリオ全体で見ると，最終目標によって示される$Impact$が最も重大である．今回の検討では，最終目標を除くフェーズの$Impact$は，無視できるほど小さいと見なす．この過程を数式で示すと，特定の攻撃シナリオで発生する影響を$Impact_{Sc}$，フェーズnにおける影響を$Impact_n$，最終目標を達成した時の影響を$Impact_{Last}$とすると，この関係は式(4.2)で表される．

$$Impact_{Sc} = Impact_0 + Impact_1 + Impact_2 + \cdots + Impact_{Last} \approx Impact_{Last} \tag{4.2}$$

今回提案する方法は，$ThreatActor$の$Goal$に基づき攻撃パターンを決定する．$AttackScenario$の$Impact$は，$AttackScenario$の$Goal$により決定される．$Goal$が一定である場合，ある特定の入口点から予想される$Goal$に至る最も可能性の高い攻撃パターンを決定することで，リスクの推定が可能になる．提案する脅威シナリオの表現モデルのうち，最も重大な脅威イベントを識別し，それを$Goal$として選択する．これは，制御システムによって重大な結果を識別し，それに基づきリスクを推定するアプローチに基づく[117]．今回はこのアプローチに基づき，$Goal$を固定とすることで，影響が重大な$AttackScenario$に絞り，本章での議論では，$Impact$を定量化する議論を省略することで，$Likelihood$の定量化に焦点を当てる．

前章示した Diamond model によると，Victim に対する攻撃成功確率は，Adversary の発現確率，Adversary の Capability に基づく攻撃成功確率，および Infrastructure から生じる攻撃活動の困難の度合いによって決定される．このモデルを踏まえると，ある脅威シナリオの$Likelihood$は式(4.3)として定式化される．

$$Likelihood = P_{ThreatActor} \times \prod_{k=0} \varphi_{C,k}\varphi_{I,k} \tag{4.3}$$

ここで，$P_{ThreatActor}$は$ThreatActor$の時間単位の発現確率を表し，$\varphi_{C,k}$は脅威アクターの攻撃能力に基づいた$AttackState_k$ $(k \in \mathbb{N})$での攻撃成功確率，$\varphi_{I,k}$は攻撃実行点の特性に依存した$AttackState_k$での攻撃成功確率を示す．

4.3. 攻撃成功確率モデルの数値的指標の議論

前節に示した$Likelihood$について，各パラメータの数値指標を議論する．$P_{ThreatActor}$は，攻撃者の発現確率であり，観測された記録に基づいて決定される．$\varphi_{C,k}$はフェーズkにおける攻撃者の能力に基づく要素であり，$ThreatActor$が特定の攻撃技術をどの程度利用できるかを意味する．$\varphi_{I,k}$は，環境上の性質に基づく攻撃難易度を意味する．例えば，同じ攻撃手法であっても，Windows マシンなどのホストデバイスと，リアルタイムオペレーティングシステム等の独自 OS を搭載した組み込みデバイスとでは，攻撃の複雑さや，成功確率が異なる．具体的な攻撃手法の例として，DoS 攻撃であるフラッド攻撃は，ホストデバイスよりも組み込みデバイスに対して効果的である．逆に，デバイス上で任意のコードを実行し，それを制御下に置き，その後さらに悪意のある行動を実行することを含む攻撃は，豊富な計算リソースと補助機能を持つホストデバイスの方が脆弱である場合が多い．したがって，同じ攻撃パターンであっても，標的システムが攻撃されやすい場合に$\varphi_{I,k}$は高い値となる．逆に標的システムが攻撃に対してあまり脆弱でない場合，$\varphi_{I,k}$は低い値をなる．

前章で説明した標的システムを例に脅威シナリオを特定し，特定の数値を使用し，$P_{ThreatActor}$，$\varphi_{C,k}$，および$\varphi_{I,k}$の各値の定義に基づき確率を計算する．最初に$P_{ThreatActor}$について，ある期間の APTxx の発現確率は 30% ($P_{ThreatActor} = 0.3$)と仮定する．次に，$\varphi_{C,k}$および$\varphi_{I,k}$を決定する．攻撃に対する脆弱性の定量化に関して様々な議論や研究が報告されているが[118][119]，その中でも最も標準化されたメトリックの一つが CVSS[120]である．本研究では，脆弱性の重大度を示す標準モデルからの値，特に CVSS Version 3 (CVSSv3)を参照する．CVSSv3 において，攻撃が悪用される可能性を意味する$Exploitability$は式(4.4)で表される．

$$Exploitability = 8.22 \times AV \times AC \times PR \times UI \tag{4.4}$$

本研究では簡略化のため，$\varphi_{C,k}$，および$\varphi_{I,k}$の評価に正規化した AC 値を利用する．具体的には，AC=High の場合，本来の High 時のスコア 0.44 を Low 時のスコア 0.77 で割った値である 0.571 を使用し，AC=Low の場合は，0.77 を 0.77 で割った 1.00 を用いる．例えば，脅威シナリオの各ステップにおいて攻撃の複雑性評価がすべて Low である場合，式(4.5)が成り立つ．

$$\varphi_{C,k\in\mathrm{N}} = \varphi_{I,k\in\mathrm{N}} = 1.00 \tag{4.5}$$

4.4. 評価

最初に，APTxx が利用可能な各 Technique の$\varphi_{C,k}$を仮定する．今回定義した表を Table 4.33 に示す．列 Conditions は各 Technique の能力が満たされる条件を指示する．例えば，T1550 (User Alternate Authentication Material)の場合，攻撃者が攻撃先の認証情報を有する（$cred_{AttackPoint} \in Cred_{ALL}$）場合，$\varphi_{C,k}$は Low となり，そうでない場合，利用可能な認証情報を探す必要があることから，$\varphi_{C,k}$は High となる．

Table 4.3 The list of $\varphi_{C,k}$ of each technique

#	Tactics	Techniques	$\varphi_{C,k}$	Conditions
1	TA0001 - Initial Access (Remote)	T1189 - Drive-by Compromise	High (0.571)	N/A
		T1133 - External Remote Services	High (0.571)	N/A
		T1566 - Phishing	Low (1.000)	N/A
2	TA0001 - Initial Access (Local)	T1199 - Trusted Relationship	Low (1.000)	N/A
		T1078 - Valid Accounts	Low (1.000)	N/A
3	TA0002 - Execution	T1059 - Command and Scripting Interpreter	High (0.571)	N/A
		T1203 - Exploitation for Client Execution	High (0.571)	N/A
		T1204 - User Execution	Low (1.000)	N/A
4	TA0007 - Discovery (Local)	T1087 - Account Discovery	Low (1.000)	N/A
		T1201 - Password Policy Discovery	Low (1.000)	N/A
		T1518 - Software Discovery	Low (1.000)	N/A
5	TA0006 - Credential Access	T1110 - Brute Force	Low (1.000)	N/A
		T1555 - Credentials from Password Stores	Low (1.000)	N/A
6	TA0007 - Discovery (Network)	T1040 - Network Sniffing	High (0.571)	N/A
		T1049 - System Network Connections Discovery	Low (1.000)	N/A
7	TA0008 - Lateral Movement	T1210 - Exploitation of Remote Services	High (0.571)	N/A
		T1021 - Remote Services	High (0.571)	N/A
		T1550 - Use Alternate Authentication Material	Low (1.000)	$cred_{AttackPoint} \in Cred_{ALL}$
			High (0.571)	N/A
8	TA0040 - Impact (Local)	T1485 - Data Destruction	Low (1.000)	N/A
		T1565 - Data Manipulation4	Low (1.000)	N/A
		T1499 - Endpoint Denial of Service	Low (1.000)	N/A
9	TA0040 - Impact (Remote)	T1565 - Data Manipulation	Low (1.000)	N/A
		T1499 - Endpoint Denial of Service	Low (1.000)	N/A

次は$\varphi_{I,k}$の数値を決定する．$\varphi_{I,k}$は，制御システムの構成上の要素である$Component$，$PhysicalArea$，および$Network$に基づいて Technique ごとに定義する．これらの値は知識ベースで決める必要がある．今回は簡単のため，攻撃の開始点が$PhysicalArea$内にある場合は High，それ以外の場合は Low として定義する．この場合，$\varphi_{I,k}$は式(4.6)となる．

$$\varphi_{I,k} = \begin{cases} 0.571, & k = 1 \ \ and \ \ lo_0 \in PhysicalArea \\ 1.000, k \geq 1 \ \ or \ \ k = 1 \ \ and \ \ lo_0 \notin PhysicalArea \end{cases} \tag{4.6}$$

上記の条件で，前節で提示された標的システムにおいて想定される全て攻撃シナリオの特定と，各シナリオの$Likelihood$を計算した．まず，可能な攻撃シナリオの総数を計算した．計算条件は，すべてのノードを開始点とし，標的の制御システムグラフから得られるすべての単純道と攻撃パターン（各攻撃パスに可能なすべての Techniques）について，攻撃シナリオを網羅探索した．以上の条件の下で，679,644 の攻撃シナリオを特定した．

その後，取得した各シナリオについて，先に述べた条件に従い，各シナリオの$Likelihood$を算出した．その結果を Figure 4.1 に示す．今回の条件では，$\varphi_{C,k}$および$\varphi_{I,k}$の定義を CVSSv3 に基づく離散モデルを利用していることから，得られる Likelihood も離散的である．今回の評価結果では，最も高い$Likelihood$のシナリオは攻撃シナリオ全体の 0.037%である結果を得た．

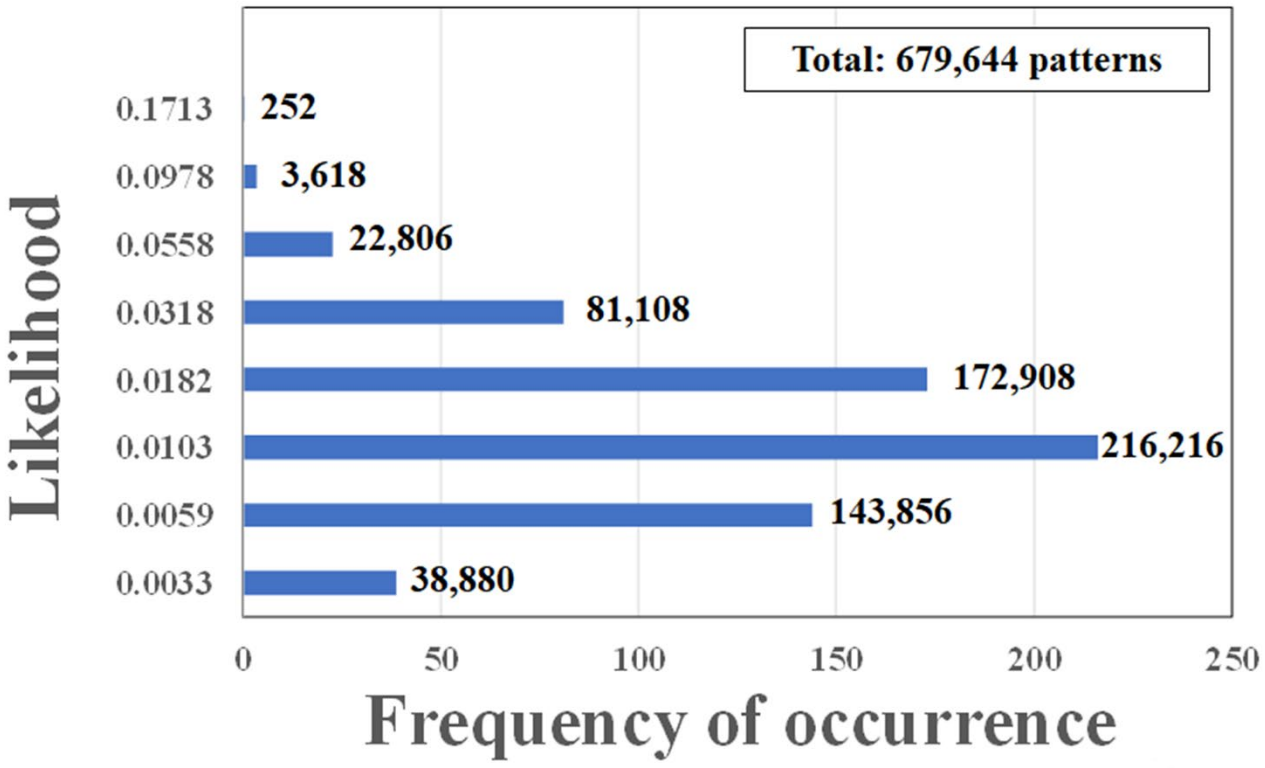


Figure 4.1 **Bar chart with the number of attack scenarios per likelihood**

次に，得られたシナリオのサンプルを Figure 4.2 に示す．Figure 4.2 は，Likelihood が 0.1713, 0.0182, 0.0059, および 0. 0033 それぞれの場合の脅威シナリオを示す．結果として，$AttackState$の数が少なく，$AttackState$が持つ$\varphi_{C,k}$および$\varphi_{I,k}$の値が高いほど，$Likelihood$が高くなる傾向が得られた．

Likelihood	Attack Pattern
0.1713	(#0 Phase:TA0001_2 Point:Internet Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1566) ➡ (#1 Phase:TA0002 Point:Gateway Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1204) ➡ (#2 Phase:TA0007_1 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1087) ➡ (#3 Phase:TA0006 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1110) ➡ (#4 Phase:TA0007_2 Point:Gateway Target:IT-NW Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1049) ➡ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1210) ➡ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1204) ➡ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1087) ➡ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1110) ➡ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1049) ➡ (#10 Phase:TA0040_2 Point:OT-NW Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1499) ➡ (END)
0.0182	(#0 Phase:TA0001_2 Point:Internet Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1133) ➡ (#1 Phase:TA0002 Point:Gateway Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1203) ➡ (#2 Phase:TA0007_1 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1087) ➡ (#3 Phase:TA0006 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1110) ➡ (#4 Phase:TA0007_2 Point:Gateway Target:IT-NW Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1040) ➡ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1550) ➡ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1204) ➡ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1201) ➡ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1110) ➡ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1049) ➡ (#10 Phase:TA0008 Point:OT-NW Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1550) ➡ (#11 Phase:TA0002 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1204) ➡ (#12 Phase:TA0040_1 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server', 'PLC'], 'cred': ['cred_1']} AttackTech:T1499) ➡ (END)
0.0059	(#0 Phase:TA0001_1 Point:Ctrl-Area Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1078) ➡ (#1 Phase:TA0002 Point:Gateway Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1203) ➡ (#2 Phase:TA0007_1 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1518) ➡ (#3 Phase:TA0006 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1555) ➡ (#4 Phase:TA0007_2 Point:Gateway Target:IT-NW Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1040) ➡ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1210) ➡ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1203) ➡ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1201) ➡ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1555) ➡ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1040) ➡ (#10 Phase:TA0008 Point:OT-NW Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1550) ➡ (#11 Phase:TA0002 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1204) ➡ (#12 Phase:TA0040_1 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server', 'PLC'], 'cred': ['cred_1']} AttackTech:T1485) ➡ (END)
0.0033	(#0 Phase:TA0001_1 Point:Ctrl-Area Target:HMI-PC Cond:{'admin': [], 'cred': []} AttackTech:T1078) ➡ (#1 Phase:TA0002 Point:HMI-PC Target:HMI-PC Cond:{'admin': [], 'cred': []} AttackTech:T1203) ➡ (#2 Phase:TA0007_1 Point:HMI-PC Target:HMI-PC Cond:{'admin': ['HMI-PC'], 'cred': []} AttackTech:T1518) ➡ (#3 Phase:TA0006 Point:HMI-PC Target:HMI-PC Cond:{'admin': ['HMI-PC'], 'cred': []} AttackTech:T1555) ➡ (#4 Phase:TA0007_2 Point:HMI-PC Target:IT-NW Cond:{'admin': ['HMI-PC'], 'cred': ['cred_1']} AttackTech:T1040) ➡ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['HMI-PC'], 'cred': ['cred_1']} AttackTech:T1021) ➡ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['HMI-PC'], 'cred': ['cred_1']} AttackTech:T1059) ➡ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1087) ➡ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1555) ➡ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1040) ➡ (#10 Phase:TA0008 Point:OT-NW Target:PLC Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1021) ➡ (#11 Phase:TA0002 Point:PLC Target:PLC Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1059) ➡ (#12 Phase:TA0040_1 Point:PLC Target:PLC Cond:{'admin': ['Server', 'HMI-PC', 'PLC'], 'cred': ['cred_1']} AttackTech:T1499) ➡ (END)

Figure 4.2 Attack scenario samples obtained in this study

4.5. 考察

前章に示した脅威シナリオの構造表現モデルに従い，$ThreatActor$による攻撃成功確率を，本手法により数値的に表現可能であることを示した．様々な$ThreatActor$と，その$Goal$の組み合わせに対して，本方式を適用することで，攻撃成功確率が最も高い脅威シナリオの特定が可能となる．これは，制御システムをモデル化したグラフのパス探索と，各フェーズでの各 Technique の$\varphi_{C,k}$，および$\varphi_{I,k}$に基づき Likelihood を計算することより得られる．また，$P_{ThreatActor}$，$\varphi_{C,k}$，および$\varphi_{I,k}$の定義は構造化された脅威シナリオモデルに関連付けられていることから，一度数値を定義できると再利用性が確保されている．

一方で，今回の結果は離散化された CVSSv3 の基準を用いており，攻撃成功確率は離散的な結果となった．より定量を得るには，$P_{ThreatActor}$，$\varphi_{C,k}$，および$\varphi_{I,k}$それぞれの値を更に定量表現する必要がある．

$P_{ThreatActor}$は，$ThreatActor$毎で異なり，実測的に決定するパラメータである．このパラメータは脅威インテリジェンスに基づき決定することができる．

$\varphi_{C,k}$を定量的に定義するには，何らかの測定を実施する必要がある．例えば，MITRE ATT&CK のようなオープンな知識データベースは，$ThreatActor$によって使用される Technique のモデルを定義しているものの，各 Technique の成功確率は定義されていない．これを定義するには，侵入検知システム(IDS)のようなセキュリティセンサを通じて$ThreatActor$の実際の攻撃を観察し，実際に使用している方法を評価するといった方法で，各 Technique に対する$\varphi_{C,k}$を決定する必要がある．この具体的な例として，セキュリティセンサを通じ，実際の$ThreatActor$が使用する技術を観測するか，または仮定された$ThreatActor$に基づいて同等の攻撃スキルを持つペネトレーションテスターによる模擬攻撃といった方法が考えられる．各 Technique の成功確率を実験的に決定することで，$\varphi_{C,k}$の更に詳細化された定義を得ることができる可能性がある．

$\varphi_{I,k}$は，ATT&CK フレームワークを参照することでその確率を，ある程度定義することができる．例えば， ATT&CK for Enterprise は，Windows や Linux など特定プラットフォームに対して共通的に適用できる Technique を提供している．ATT&CK を参照することで，Technique がプラットフォームに対応している場合，その Technique は利用可能（$\varphi_{I,k}=1$）と判断できる．その一方，$\varphi_{C,k}$の議論と同様，実際の成功確率を定量的に定義することは困難であり，公理に基づく数理的なモデル定義が困難である場合，観測，または実験的に決定する必要がある．以上より，$\varphi_{C,k}$，および$\varphi_{I,k}$の更なる定量化は残課題である．

本論の議論で割愛した$Impact$の定量表現も課題として残る．基本的なアプローチは，$Impact$の波及効果を安全性分析手法や事業影響分析手法を用いて評価し，曖昧な要素を排除することが不可欠である．$Impact$の定量表現に関する関連研究がいくつか報告されている[121][122]ものの，$Impact$を論理的な表現モデルは不明確な部分が多い．例えば，サイバー攻撃による$Impact$の伝播をモデル化する手法[113]，金銭単位でモデル化する手法[60]などがある．また，報告者を含む研究チームにより，制御システム業務の情報モデルを定義し，そのモデルに基づき，攻撃を受けた結果の伝播モデルに関する報告をしている[61]．一方で実用的なモデルや手法は，依然確立されておらず，引き続き研究の余地がある．

最後に，リスク評価過程におけるプロセス効率化の効果について述べる．本提案手法により，$Likelihood$の属人性の排除を部分的に達成した．一方で，$P_{ThreatActor}$，$\varphi_{C,k}$，および$\varphi_{I,k}$それぞれを完全な属人性排除しつつ，実験的，または第一原理的に決定する手法を確立するまでは，属人的に決定する必要がある．各パラメータの客観的な決定手法の確立は，今後の課題である．また，$Impact$に関しては，上述の通り，本提案手法では属人性の排除，または自動化は達成できないため，その手法の確立は今後の課題である．

Table 4.4　Achievement status of elimination of individuality or automation by the proposed method

#	Process	Description	Achievement
1	**Estimate degree of impact**	Estimate the degree of impact caused by a cyber attack	None
2	**Estimate degree of likelihood**	Estimate the degree of likelihood which the threat is likely to materialize	Partial

4.6.　本章のまとめ

制御システムの脅威シナリオの優先順位付けにあたり，リスクの評価基準が定量的であるほど，客観的な評価結果を得られるが，定量的な評価基準の設計と，その評価プロセスが定性的な基準と比べて複雑である．また，既存手法では，リスク値の見積もりにおいて攻撃者の特性に起因する要素の考慮が不十分である課題があった．

この課題に対し，前章で示した脅威シナリオの構造化表現モデルを基にリスク推定のための定量的指標を導入し，攻撃者の特性を考慮した脅威シナリオの優先順位付けにおける主観的判断を排除す手法を提案した．この手法は，前章で示す脅威シナリオの内部モデルに基づき，オントロジに基づき要素分解した各攻撃フェーズの攻撃成功確率を決定する要因をモデル化し，リスクを可能な範囲で定量化表現することで，属人性を排除することが狙いである．

本手法は，攻撃者の顕在化確率，攻撃者の能力に基づく攻撃の成功確率，および攻撃実行時の環境に基づく成功確率からなる確率モデルとして脅威シナリオの攻撃の成功確率をモデル化する．本モデルに基づき，制御システムにおける潜在的な侵入点から，最終目標に至るまでの確率を算出することで，脅威シナリオ全体の成功確率を導出する．その結果，到達する可能性がある脅威シナリオの中で優先順位を，定量的なリスク値に基づき決定できる，更に各パラメータは再利用性が確保されていることを示した．

第5章 リスク対処：キルチェーンと多層防御に基づく対策設計方式

5.1. はじめに

前章までに，脅威シナリオの表現モデルと，当モデルに基づくリスク評価手法を述べた．本章では，オントロジ駆動型モデリングに基づくリスクに基づくセキュリティリスク対処の効率化，自動化を実現する手法について詳説する．

制御システムのセキュリティ仕様を得るアプローチとして，大きくベースライン型とリスク解析型がある[126][127]．ベースライン型は，業界や組織で標準化されたセキュリティ対策のセットに合わせ立案する．ベースライン型はリスク識別，リスク評価プロセスを省略することから，工数上で有利あるが，多数存在するセキュリティ対策の内，優先的に実践すべき対策を判定できない点や，対策が過剰になる恐れがあるといった課題がある．逆にリスク解析型は，制御システムのセキュリティ脅威を識別し，識別したリスクを評価し，当該リスクを低減，回避するセキュリティ対策を決定する．つまり，リスク解析型は脅威モデリングそのものである．リスク解析型は，リスクを明確にしたうえで優先すべきセキュリティ対策を決定することから，費用対効果の高いセキュリティ対策を明確化する点で有用である反面，その実施に係る工数が課題である．

上記課題に対処するにあたり，前章までに，オントロジ駆動型モデリングに基づく脅威モデリングにおけるリスク識別，リスク評価それぞれの評価手法を示した．その一方で，効果的なセキュリティ対策を決定する手法の研究はリスク識別，リスク評価の手法に関する研究と比べ，多く報告されていない．制御システムオーナにとっては，セキュリティ対策を決定する方が重要である．

セキュリティ対策の設計手法に関する既存研究の例をいくつか挙げる．W. Wideł らが提唱する対策設計法は，与えられた攻撃ツリーにおいて，攻撃者が攻撃目標に達成するまでの時間(TTC: Time To Compromise)をシミュレーションする方式であり，TTC を計算する上で，対策の抑止効果のモデルを導入する[60]．考え得るセキュリティ対策パターンを挙げ，そのパターン毎の TTC を計算し，最も TTC が長く，かつセキュリティ対策のコストがかからない対策を決定する手法である．他の例として，O. Stan らが提唱する対策設計法は，攻撃ツリーで表現された脅威シナリオのリスク値に対して，対策によるリスク値の変化量を数学的なモデルによって表現する方式を提案している[61]．この方式は，リスク値の変化モデルと，実践コストを含む対策モデルの一覧を元に，ヒューリスティック探索でリスク値の変化量を計算し，最もコスト効果の高い対策を決定する．また，L. Wang らから提案された手法は，攻撃グラフとセキュリティ対策効果のメトリクス値を割り当てることで，システム内のネットワーク全体でセキュリティ対策を最適化するフレームワークが提案されている[62]．最後に，Y. Fei らの先行研究も同様，攻撃ツリーに対して，防御ツリーを構築し，ツリーの各リーフに対して，定量的なセキュリティに関するメタモデルを付与することで，セキュリティ対策効果を定量化するフレームワークを提案している[63]．

これら手法は，いずれもオントロジに基づき，事前にモデル化された多数のセキュリティ対策を試行錯誤的に組み合わせ，与えられた予算内で最大のセキュリティ効果を得る対策のパターンを見つけ出すものである．

一方，上記の方式は，事前に利用できるセキュリティ対策のモデルが既知であり，更に対策に係るコストや，対策のリスク低減効果モデルも既知でなければならない．言い換えると，利用できるセキュリティ

対策のモデルが未知の場合は，これら既存の手法では優先的なセキュリティ対策を決定できない．この場合，既知のセキュリティ対策ありきでの対策立案となってしまい，セキュリティ対策手法に関する知識が必要になり，更にセキュリティ対策提供事業者の利害関係が存在する場合，セキュリティ対策の選定が恣意的になる恐れがある．この問題を回避するには，セキュリティ対策が未定な状況においても，各対策の優先順位を決定する必要がある．本研究では，そのような状況であっても，脅威シナリオに基づき，費用に対して効果が高い対策を機械的に獲得する方法の確立が目的である．これを実現するため，前章までに示したオントロジ駆動型モデリングに基づき表現された脅威シナリオを活用し，セキュリティ対策の決定要因のオントロジを明らかにし，そのオントロジに基づくセキュリティ対策の優先順位決定アルゴリズムを開発した．次節以後，その設計と評価結果を述べる．

5.2. 基本モデル

5.2.1 設計方針

前章に示した通り，リスクは式(5.1)で表すことができる．

$$Risk = Impact \times Likelihood \tag{3.1}$$

リスクを決定する手法として，多く活用されているのが，Figure 5.1 に示すリスクマトリックスである[132][128]．この例では，横軸に脅威事象の影響度，縦軸に脅威事象が送りうる可能性を示し，影響度と可能性の組み合わせでリスクの度合いを決定する．

情報セキュリティにおいて，影響度および可能性の共通的かつ定量的な指標・尺度は確立されておらず，多くは定性的な指標に基づいて決定される．たとえば，ISO/IEC 27005 では，5 段階のレベルが定義される[128].

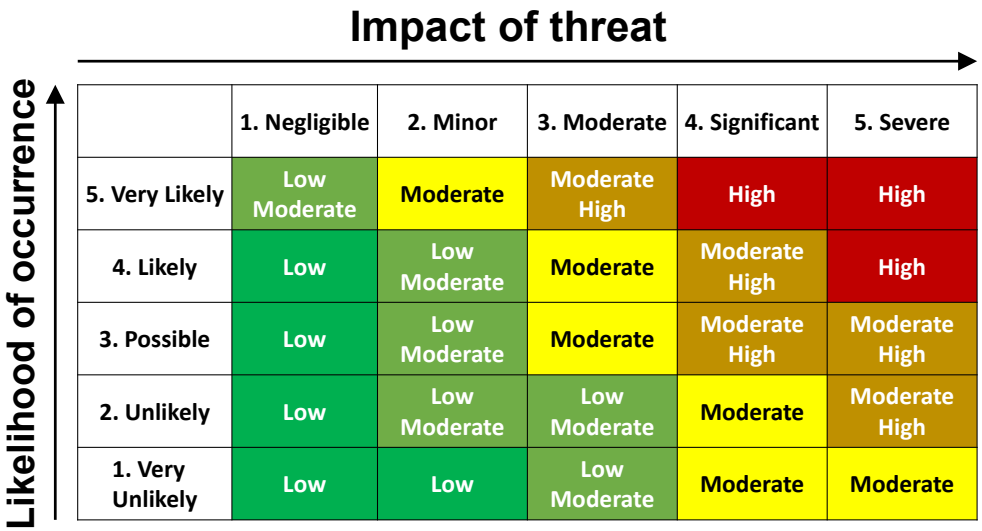


Figure 5.1 An example of the risk matrix

リスク管理において，対策の十分性は，ある脅威事象のリスクを，セキュリティ対策により，許容範囲内に抑えることが可能かどうかで決定する．言い換えると，対策により脅威が発生する可能性の低減，または脅威が現実化した場合の影響の低減により，リスクが許容範囲である場合，対策が十分とみなす．

Figure 5.2 にリスクマトリックスを使って効果を視覚的に表現した例を示す．リスクの定義に基づくセキュリティ対策効果とは，「セキュリティ侵害に起因する脅威事象のリスク（影響度，および可能性）を低減する度合い」を意味する．

例えば，ある脅威事象が起こる Likelihood が当初"Very Likely"であったとした場合，対策 X により，その Likelihood が"Very Unlikely"まで軽減できたとした場合，この変化量を「対策 X の効果」とする[130]．この対策効果の表現方法は，直感的に効果を可視化できる点で有用である．その一方，対策毎の変化量の判定は，基準が明確でなければ判断者の主観に依存する傾向がある．この課題を解決するには，対策効果を定量的に表現するモデルが必要である．

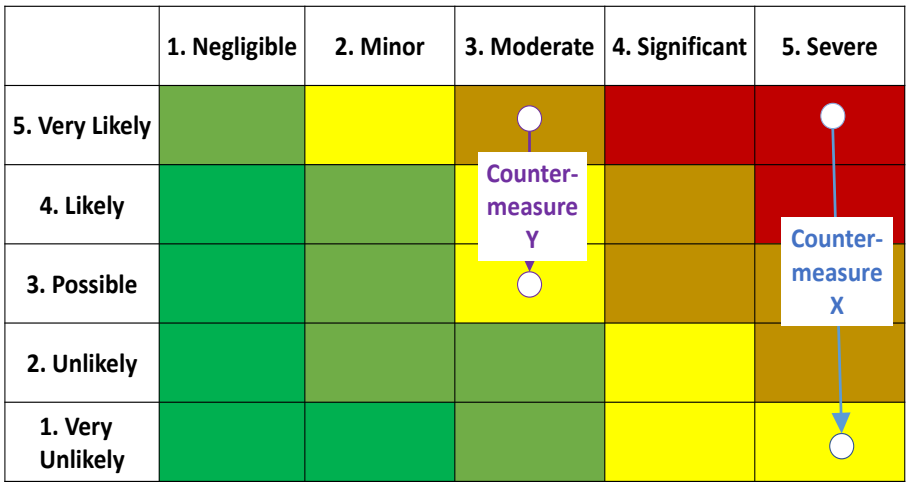


Figure 5.2　The visual image of countermeasures effects

いずれの既存研究も，独創的なアプローチで対策効果を定量的に表現するモデル構築を試みている．これら既存の手法は，属人性排除の観点で有用である反面，事前にセキュリティ対策の効果のモデルが定義されている必要がある．それに加えて，対策モデルが具体的，かつ定量的である程，特にセキュリティ対策に関する知識が不十分な場合に，対策モデルの事前定義が困難化する．

この課題の解決に向け，オントロジに基づき，セキュリティ対策の決定モデルを検討し，セキュリティ対策が不確定であっても活用可能な，セキュリティ対策決定モデルを検討した．

5.2.2　問題設定

問題の基本は，制御システムの脅威シナリオに基づき優先するセキュリティ対策を求めることである．これは脅威モデリングの基本的な目的である[131][132]．この問題を具体的に設定するにあたり，脅威シナリオのモデルを明確にし，その脅威シナリオからセキュリティ対策を決定するモデルを定義する必要がある．セキュリティ対策の戦略モデルはいくつか存在するが，今回，サイバーキルチェーンモデルと多層防御モデルに着眼する．

最初にサイバーキルチェーンモデルについて述べる．脅威シナリオは，制御システム上のある侵入点から，最終的に攻撃者が意図する最終的な資産に向かい，一連の攻撃を実行する．一連の攻撃すべてが成功すると，脅威が現実となる．これを防御するには，セキュリティ対策により一連の攻撃のどこかで，それを阻止する．具体例を挙げて説明すると，例えば，システム内部に侵入するため，最初に制御システムユーザに対して Phishing 攻撃を実施する．その後，その制御システムユーザが利用するオペレータコンソールを乗っ取り，それ経由でシステム内部へ侵入を試みる．途中認証機能がある場合，それを迂回する攻撃を実行する．これらのような攻撃が連続的に成立し，最終的に脅威が実現される．

逆に言えば，一連の攻撃活動のどこかを対策し，攻撃者の活動を阻止できれば，脅威の現実化を回避できる．攻撃者の一連の攻撃活動のどこかを切断することで，攻撃を防御する概念は，Figure 5.33 に示すサイバーキルチェーンモデル[87]と呼ばれる．

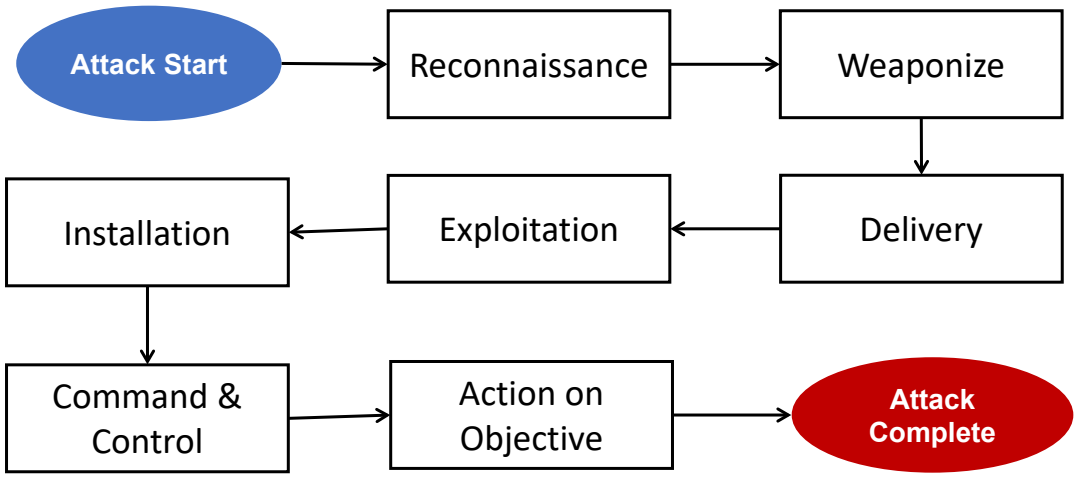


Figure 5.3　Cyber kill chain model

本論で提案する手法は，二つの手順からなる．最初に前章まで示した脅威シナリオの表現モデルを用いて，脅威シナリオを，サイバーキルチェーンモデルに基づくモデルで表現する．次に，表現したサイバーキルチェーンモデルに基づくセキュリティ対策モデルを立案する．第 3 章に示した脅威表現のモデルを活用した場合，脅威シナリオにおける一連の攻撃アクション（攻撃シナリオ）を$Killchain$とし，ステップkの攻撃状態を$AttackState_k$とした場合，$Killchain$を数学的に表現すると，式(5.1)となる．ここでk, nは，$\{k, n \in \mathbb{N} \mid 0 \leq k \leq n\}$である．

$$Killchain = AttackState_1 \prec AttackState_2 \prec AttackState_3 \prec \cdots \prec AttackState_k \prec \cdots \prec AttackState_n \quad (5.1)$$

$AttackState_k$は，式(5.2)に示される順序組の形式で表現される．p_kは攻撃フェーズ，At_kは攻撃技術・手法を要素とする有限集合，$l_{from,k}$はp_kにおける攻撃者の位置，$l_{to,k}$はp_kにおける攻撃先要素である．

$$AttackState_k = (p_k, At_k, l_{from,k}, l_{to,k}) \tag{5.2}$$

すなわち，$Killchain$は，$AttackScenario$と同一である．本章に限り，攻撃シナリオ($AttackScenario$)をキルチェーン($Killchain$)という表現で統一する．次に，セキュリティ対策を決定するモデルを説明する．$Killchain$を構成する一連の攻撃状態の集合$AttackState_1, AttackState_2, \ldots, AttackState_n$に対し，どれかの攻撃状態の成功を妨害することで，攻撃者が最終目標の到達を阻止する．

例えば，$0 \leq k, m \leq n$とした場合，セキュリティ対策M_1により，$AttackState_k$の成功を阻止できれば，$Killchain$の達成を阻止できる．更にセキュリティ対策M_2により，$AttackState_m$の成功を阻止した場合，セキュリティ対策M_1, M_2により，$Killchain$に対して 2 点の防御ポイントができる．このポイントを本論では防御層(Defense-layer)と呼ぶと，$Killchain$上の防御層が増える程，攻撃者が最終目標を達成する確率を更に低減できる．この概念は多層防御と呼ばれるものである．

多層防御(Defense-in-Depth)とは，最終的に守る対象に対して，複数の防御層を確保することで，防御を強靭化するセキュリティモデルである[9][133]．そのイメージを Figure 5.44 に示す．防御層の効果の違いが無いと仮定した場合，防御層の数が増えるほど，攻撃者が最終的な目標を達成する可能性を低減できるため，防御層を多くするほど，$Killchain$に対してセキュアになることを直感的に理解できる．

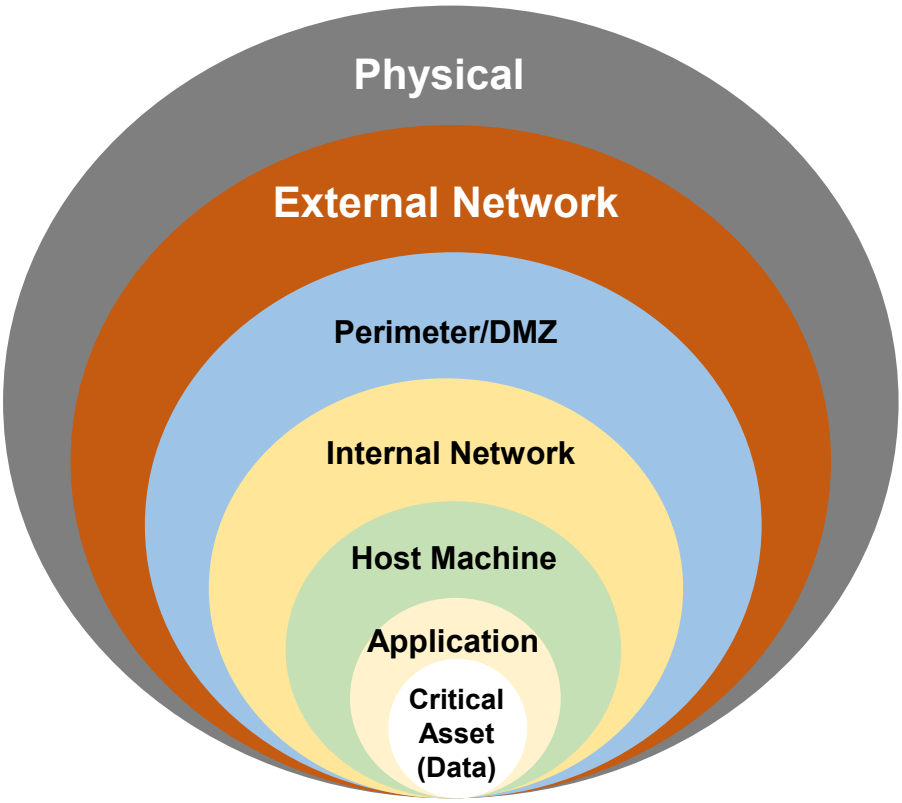


Figure 5.4　Defense-in-Depth model

一つの制御システムを見ても，攻撃パスは多数存在する．つまり，制御システムからは複数の$Killchain$が得られる．この状況において，理想的なセキュリティ対策は，「少ない対策で多くの$Killchain$に対処す

る」効果を得られるものである．例えば，対策 A と対策 B について，$n > m$とした場合，対策 A はn個の$Killchain$に対処でき，対策 B はm個の$Killchain$に対処できるとした場合，対策 A の方が優先する対策と結論付けられる．この考えを応用すると，セキュリティ対策が未定な状況であっても，セキュリティ対策の優先順位を決定できる．

次に，セキュリティ対策を実施する$AttackState$の優先順位を決定する．攻撃者は，システム上で想定される複数の$AttackState$を経て$Killchain$を遂行する．前述の通り，$Killchain$は複数パターン存在し得るが，中には，異なる$Killchain$間で共通する$AttackState$が存在する場合もある．例えば，それぞれ異なるインスタンスとして，$Killchain_i$と$Killchain_j$とし，それぞれ式(5.3)，(5.4)で表した場合，各$Killchain$に共通する$AttackState$が存在する条件は式(5.5)となる．$Killchain_i$と$Killchain_j$は，実際は式(5.1)に示す有向集合である．ここで，$AttackState_{p,q}$は，$Killchain_p$におけるq番目の攻撃状態である．

$$Killchain_i = \{AttackState_{i,k} | 0 \leq k \leq n\} \tag{5.3}$$

$$Killchain_j = \{AttackState_{j,m} | 0 \leq m \leq n\} \tag{5.4}$$

$$AttackState_{i,k} = AttackState_{j,m} \tag{5.5}$$

式(5)を見たす$Killchain_i$と$Killchain_j$の攻撃状態の集合は，$Killchain_i \cap Killchain_j$で表され，式(5.6)を満たす．

$$Killchain_i \cap Killchain_j \neq \emptyset \tag{5.5}$$

式(5.6)を満たす場合，各$Killchain$の攻撃パス上に，共通する$AttackState$が必ず存在する．上記の基，解く問題は，攻撃対象とする制御システムにおいて識別されたリスクが高い$Killchain$に対し，最も多くの$Killchain$に重複する$AttackState$を抽出することである．この問題は，以下で解くことができる．

(1) 対処すべき（高リスクの）$Killchain$の全組み合わせパターンを求める．
(2) (1)で求めた$Killchain$に対し，共通する$AttackState$が存在する$Killchain$を抽出する．
(3) (2)で抽出した$Killchain$の内，最も多い$Killchain$に含まれる$AttackState$を決定する．

次に上記を定式化する．$AttackState$を元，重複の無いNパターンの$AttackState$の有限集合$Killchain$から構成される有限集合族とした場合，$Killchain$の全組み合わせは，それらの冪集合族となる．$Killchain$から構成される有限集合族を$\{Killchain_k\}_{k=1}^{N}$，この集合族の要素を$a$と置くと，$a$は式(5.6)を満たす．

$$a \subset \{Killchain_k\}_{k=1}^{N} \tag{5.6}$$

ここで，aに属する全$Killchain$に共通の$AttackState$が存在する条件は式(5.7)で表される．

$$\bigcap a \mid a \in \mathcal{P}(\{Killchain_k\}_{k=1}^{N}) \neq \emptyset \tag{5.7}$$

式(5.7)を満たす，要素aから成る集合を式(5.8)とする．この場合，A_Cは，$A_C \subset \mathcal{P}\{Killchain_k\}_{k=1}^{N}$を満たす．

$$A_C = \{a | \bigcap a \neq \emptyset\} \tag{5.8}$$

A_Cの要素の内，最も多くの$Killchain$に含まれる要素をa_Mと表すと，a_Mは式(5.9)を満足する集合族である．

$$a_M | a_M \in A_C, a_M \subset \{Killchain_k\}_{k=1}^{N} \tag{5.9}$$

a_Mは明らかに式(5.8)を満足する．したがって，a_Mを構成する全要素（$Killchain$）に共通する$AttackState$が必ず存在する．この$AttackState$を要素とする集合をPと置くと，Pは式(5.10)で示される．Pの要素sが，最も優先すべき$AttackState$となる．

$$P = \{s | s \subset \bigcap a_M\} \tag{5.10}$$

次に解く問題は，Pを計算し，その結果から$AttackState$の優先順位を決定することである．次節以降，本問題の解法を述べる．

5.2.3 主要モデルの定義

A. キルチェーンモデル

サイバーキルチェーン（以後，キルチェーン）は，始点となる攻撃状態と，終点となる攻撃状態の両方が必ず存在する．攻撃者は，始点から攻撃活動を開始し，途中で様々な攻撃状態を経て，攻撃の目標を達成するか，攻撃が途中で防御，または検知される，もしくは攻撃者の能力不足により攻撃が失敗するまで試行する．$AttackState$を用いてこれを図示すると，Figure 5.55 に示す状態機械として表現する．$Killchain$は，式(5.1)で表すモデルであり $AttackState_k$は式(5.2)に示されたモデルである．

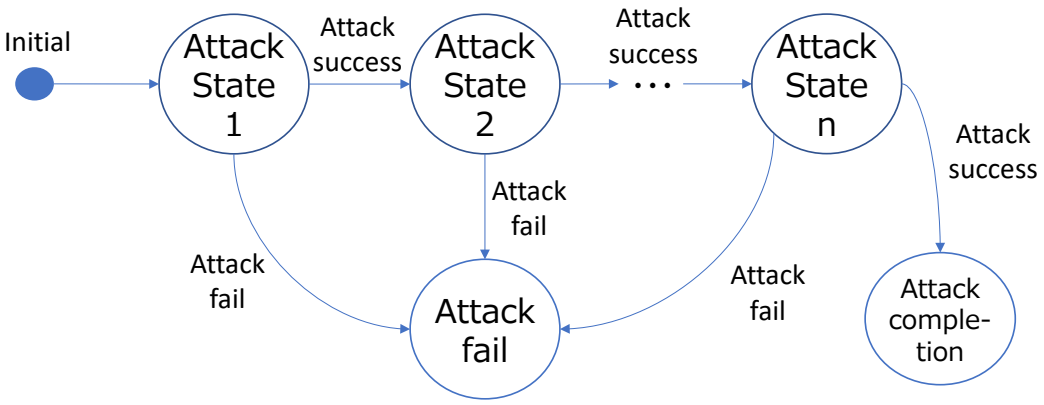


Figure 5.5　A state transition model of an adversary

$AttackState_k = (p_k, At_k, l_{from,k}, l_{to,k})$において，攻撃フェーズ$p_k$は，攻撃者における部分目標である．例えば，Initial Access（初期侵入），Lateral Movement（横展開），Collection（情報収集），Privilege Escalation（権限昇格）といった攻撃戦術が例である．これらは，攻撃フェーズは知識ベースで定義され，そのデータセットに依存する．データセットの例として，MITRE ATT&CK Tactics がある．

ATT&CK Tactics を例にp_kを定義すると，ATT&CK Tactics の要素から成る有限集合を$Tactics$とし，$tac_1, tac_2, \dots tac_x$をその要素とすると，$p_k$は式(5.11)を満たす．

$$p_k \in Tactics \mid Tactics = \{tac_1, tac_2, \dots, tac_x\} \tag{5.11}$$

At_kは，p_kを達成するための攻撃手法を要素とする有限集合である．攻撃手法は，例えば DoS 攻撃や，リスト型攻撃といった個々の攻撃手法が該当する．攻撃者が利用可能な全攻撃手法$t_0, t_1, t_2, \dots$の有限集合をAt_{adv}=$\{t_0, t_1, t_2, \dots\}$とすると，At_kは，At_{adv}の部分集合($At_k \subseteq At_{adv}$)を満たす．このAt_{adv}もp_kと同様，知識のデータセットから決定される．データセットの例として，MITRE ATT&CK Techniques がある．ATT&CK Techniques の有限集合を$Techniques$とし，その要素を$teq_1, teq_2, \dots teq_x$とすると，$At_{adv}$は式(5.12)を満たす．

$$At_{adv} \subseteq Techniques \mid Techniques = \{teq_1, teq_2, \dots teq_x\} \tag{5.12}$$

At_{adv}の要素数が多いほど，多くの攻撃手法を有することとなることから，攻撃が成功する確率は高くなる．$AttackState_k$において，At_kの各要素となる攻撃技術を試行し，攻撃が成功した場合は，$AttackState_{k+1}$に遷移する．逆に，$At_k = \emptyset$であるか，$At_k \neq \emptyset$であっても，システム側で，At_kに対する対策がなされている場合は，攻撃失敗となる．

また，攻撃者の現在位置$l_{from,k}$，攻撃先$l_{to,k}$は，システム上の点であるため，これを実体化するには，攻撃対象システムのモデルが必要となる．

B.　システムモデル

次に攻撃対象である制御システムのモデルについて述べる．これは，第 3 章に述べたモデルを利用する．攻撃対象システムを$TargetSystem$とすると，その要素は式(5.13)で示される．

$$TargetSystem = PhysicalArea \cap Component \cap Network \tag{5.13}$$

$PhysicalArea$は，システムを構成する物理的な空間要素の有限集合であり，システムを構成する物理的な空間要素を$phy_1, phy_2, \dots, phy_x$とすると$PhysicalArea$は式(5.14)で示される．

$$PhysicalArea = \{phy_1, phy_2, \dots, phy_x\} \tag{5.14}$$

$Component$は，システム内に存在するコンポーネントの有限集合である．コンポーネントの要素を$comp_1, comp_2, \dots, comp_3$とすると，$Component$は，式(5.15)となる．

$$Component = \{comp_1, comp_2, \dots, comp_3\} \tag{5.15}$$

$Network$は，システム内に存在する論理ネットワークの有限集合である．論理ネットワークの要素を$nw_1, nw_2, \dots, nw_3$とすると，$Network$は，式(5.16)となる．

$$Network = \{nw_1, nw_2, \dots, nw_3\} \tag{5.16}$$

$TargetSystem$の各構成要素の関係モデルを ERD で表すと Figure 5.66 になる．この表記は Crow's foot database notation[134]に基づき各モデル間の関係を定義している．

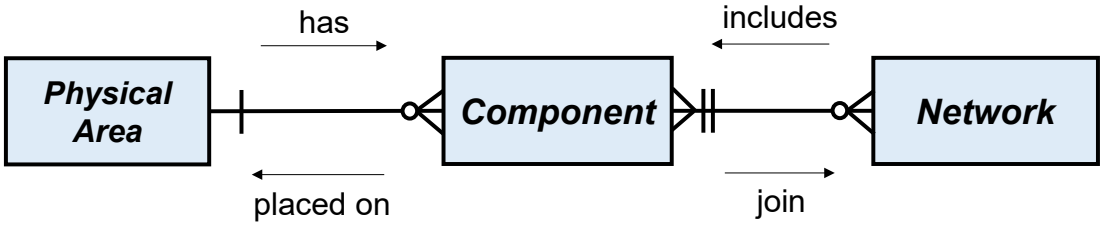


Figure 5.6 The E-R model of the $\boldsymbol{TargetSystem}$

このモデルは，$Network$は，2 台以上の$Component$から構成され，$Component$は，0 個の$Network$に参加する．$Component$は単一の$PhysicalArea$に配置され，一つの$PhysicalArea$は 0 個以上の$Component$が存在することを説明する．このモデルは，攻撃成立可否を判断する要素となる．例えば，phy_1から$comp_1$に対する直接的な攻撃は，関連が存在することから成立するが，phy_1からnw_1への直接的な攻撃は，関連が存在しないため成立しない．つまり，phy_1からnw_1への攻撃は，$comp_1$を介する必要があり，攻撃経路は結果的に$phy_1, comp_1, nw_1$となる．これは，Figure 5.66 に示すモデルを満たす．

このモデル元に，$TargetSystem$の攻撃パスを$TargetSystem$の各要素をノードとするグラフで表現する．その例を Figure 5.77 に示す．

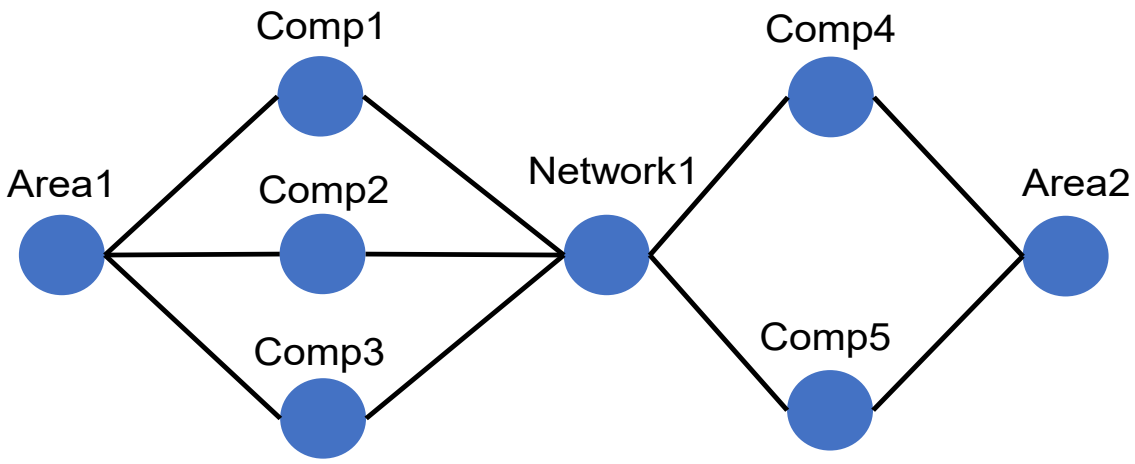


Figure 5.7　A graph image of the $\boldsymbol{TargetSystem}$

本グラフにおけるエッジは，$Component$，または$PhysicalArea$であり，物理アクセス，または論理アクセスが可能なポイントである．これを数学的に表現すると，式(5.17)-(5.19)となる．

$$Graph = \{V, E\} \tag{5.17}$$

$$V = \{Area1, Area2, Comp1, Comp2, Comp3, Comp4, Comp5, Network1\} \tag{5.18}$$

$$E = \{(Area1, Comp1), (Area1, Comp2), (Area1, Comp3), (Comp1, Network1), \dots\} \tag{5.19}$$

攻撃者は，$TargetSystem$内のいずれかの要素に存在し，任意の要素を攻撃対象とし，アクションを実行する．l_{from}，l_{to}は，$TargetSystem$上の任意点であり，式(5.20)で表現される．

$$l_{from}, l_{to} \in TargetSystem \tag{5.20}$$

節 5.2.2 に設定した問題に対する解法アルゴリズムを次節で詳説する．

5.2.4　優先順位決定アルゴリズム

A.　キルチェーン一覧の獲得

前節に示したグラフを基に，始点と終点を定め，グラフ上で可能なパスを導出することでキルチェーンを獲得できる．一方，始点と終点の組み合わせとパスに制約条件が無ければ，組み合わせ爆発により，考え得るキルチェーン数が膨大になる可能性がある．一方で，始点と終点は，それぞれリスク値が異なる．例えば始点は攻撃者にとって起点となりやすいポイントの方が，高いリスクとなる．例えば，外部ネットワーク等のシステム外の論理ネットワークが高リスクの始点の例である．終点も影響が大きいコンポーネントの方が，高リスクである．したがって，リスクが高い重要なキルチェーンに絞って抽出するために

は，優先的に考慮すべき始点，終点のノードを決定し，そのうえで，パスを列挙する必要がある．以下，重要なキルチェーンを獲得する手順を詳説する．

（1） 始点，終点の組み合わせを決定する

キルチェーンの始点と終点を事前分析により決定する．この方式は，橋本らに報告された制御システムの悪影響を中心とした対策策定手法に基づく[136]．この事前分析では，始点ノードと，終点となりうるノードを，システム構成と，システムの各コンポーネントのアプリケーション上の役割に基づき決定する．例えば，誰でもアクセス可能な物理エリア，もしくが，外部の論理ネットワークなど，起点となる可能性が高いポイントを抽出する．

終点は，攻撃が達成されたことにより，人命への影響や環境への悪影響，事業停止などの制御システムの業務への影響など，制御システムオーナにとって悪影響に繋がる恐れのあるポイントを選ぶ．これを決定する手法として，FTA (Fault-Tree Analysis)や，FMEA (Failure Mode and Effects Analysis)などの手法[137]を活用することができる．始点，終点の抽出後は，それらの組み合わせを列挙し，対策を講じる組み合わせを抽出する．

（2） 始点－終点間の攻撃パスを決定する

始点と終点の組み合わせが決まった後は，可能性のある攻撃パスを列挙する．攻撃パスの列挙では，エッジの通過は一度迄とする．すなわち，パスは必ず単純道でなければならない．Figure 5.77 を例に， Area1 を始点，Comp5 を終点とすると，全部で 4 パターンのパスが得られる．

（3） 攻撃パスの攻撃状態を決定する

（2）で決定した攻撃パスは攻撃者の位置情報でしかない．攻撃パスをキルチェーンとして表現するには，攻撃パス中の各点に対して，攻撃フェーズと攻撃手法を関連付けることで，$AttackState$を定義する．

B. 優先順位の決定

次に，キルチェーンからセキュリティ対策を決定する．最も理想的な状態は，キルチェーンの全ての攻撃状態に対して，防御層が形成されていることである．一方，この理想形は，コスト上の理由で多くのシステムに実践することは困難でり，防御層の導入箇所の優先順位を決定する必要がある．

先ず，防御層を配置する基本戦略を述べる．対策が必要なキルチェーンに対し，防御層は以下の観点を踏まえて配置箇所を決定する．

i. 可能な限り最小の対策で，可能な限り多くのキルチェーンを防御する
ii. 可能な限りキルチェーンの早期フェーズで攻撃を防御する

上記 i は，式(5.10)の集合Pの計算に対応し，ii は可能な限り攻撃を早期に検知し対処したいという普遍的な防御側の心理に対応する．これらに基づき，$TargetSystem$から識別し，重要なキルチェーンに対し，防御層の配置箇所の優先順位を決定するアルゴリズムを検討した．設計したアルゴリズム仕様の詳細を以下に述べる．

(1) $TargetSystem$から識別したキルチェーンのうち，防御対象となるキルチェーン（防御対象キルチェーン）を選択する．

(2) N 番目$(1 \geq N \geq maxlength)$の防御層を全てのキルチェーンに対して形成を試みる．各キルチェーンの構成する攻撃状態のうち，最も多くのキルチェーンに含まれる攻撃状態を抽出する．このポイントを攻撃状態 X と置くと，攻撃状態 X が，N 番目の防御層形成時の最優先の防御層形成ポイントとなる．尚，この$maxlength$は，防御形成対象キルチェーンのうち，攻撃状態の最大数である．

(3) 次に，攻撃状態 X を含まないキルチェーンを抽出し，それらの中で最も多く共通する攻撃状態を，攻撃状態 X の次に N 番目の防御層形成時の優先的なポイントとなる．

(4) 全てのキルチェーンでN番目の防御層が形成されるまで，上記の処理を繰り返す．

(5) (2)-(4)を$N = 1$から，$N = maxlength$まで繰り返す．この処理により，全てのキルチェーンの全ての攻撃状態に対して優先順位を決定できる．

以上に示した仕様のイメージを Figure 5.8 に示す．簡単のため，システム上のポイントのみ挙げる．例えば，与えられたキルチェーンが 8 本の場合，最も重複するポイントは，Component A であることから，最も優先度が高いポイントは Component A となる．次に重複が多いポイントは Network B, および Network C である．この場合，次に優先度が高いポイントはこれら二つのネットワークとなる．以上をすべてのポイントに対して繰り返すことで，優先順位を得る．

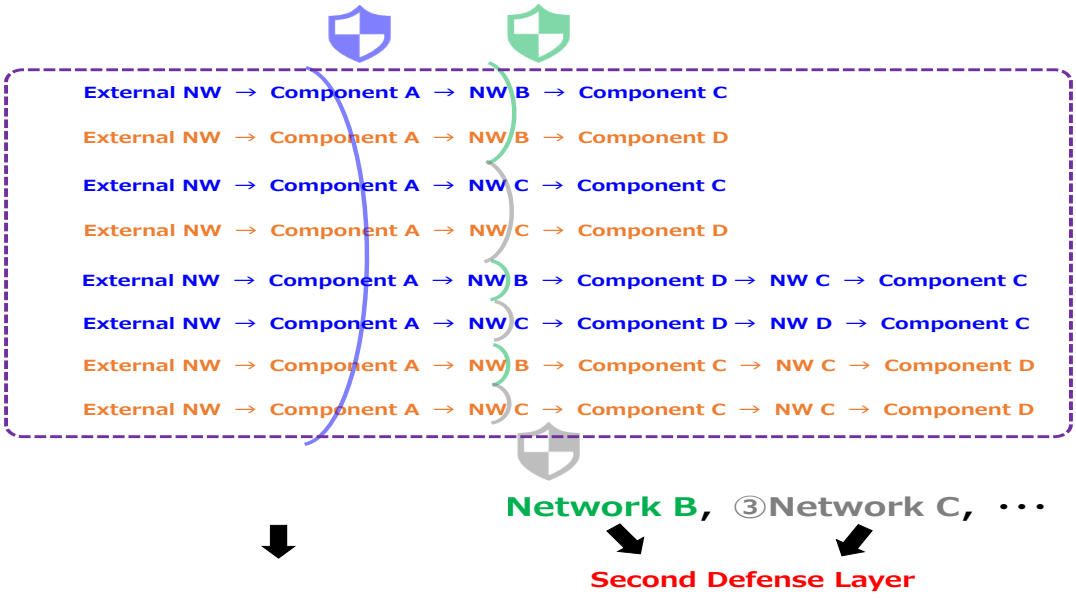


Figure 5.8　A conceptual image of the prioritization

Figure 5.9 に上記を踏まえた上で設計したアルゴリズムを示す．以後文章内の"行 nn"は Figure 5.9 に示すアルゴリズムの行番号を意味する（nn は二桁の整数）．

The algorithm that prioritize defense layer points

01: List up all possible kill chains (**KillChain$_{poss}$**)
02: Select kill chains considered for defense layers (**KillChain$_{def}$**)
03: Determine the maximum kill chain length in **KillChain$_{def}$** (maxlength)
04: N ← 1
05: **checkedAttackStates** ← ∅
06: priority ← 1
07: **while** N ≤ maxlength **do**
08: **KillChain$_{undef}$** ← **KillChain$_{def}$**
09: Exclude kill chains that have already defense layers greater than or equal to N from **KillChain$_{undef}$**
10: **while KillChain$_{undef}$** ≠ ∅ **do**
11: Determine the attack state sets the most appeared in **KillChain$_{undef}$** that is not included in **checkedAttackStates** (**a$_{most}$**)
12: Output **a$_{most}$** and priority
13: Determine the number of **a$_{most}$** (Num(**a$_{most}$**))
14: priority ← priority + Num(**a$_{most}$**)
15: Determine kill chain(s) that contain any element of **a$_{most}$** (**KillChain$_{nocont}$**)
16: **KillChain$_{undef}$** ← **KillChain$_{undef}$** \ **KillChain$_{nocont}$**
17: **checkedAttackState** ← **checkedAttackState** ∪ **a$_{most}$**
18: **end while**
19: N ← N + 1
20: **end while**

Figure 5.9 The algorithm of prioritizing points

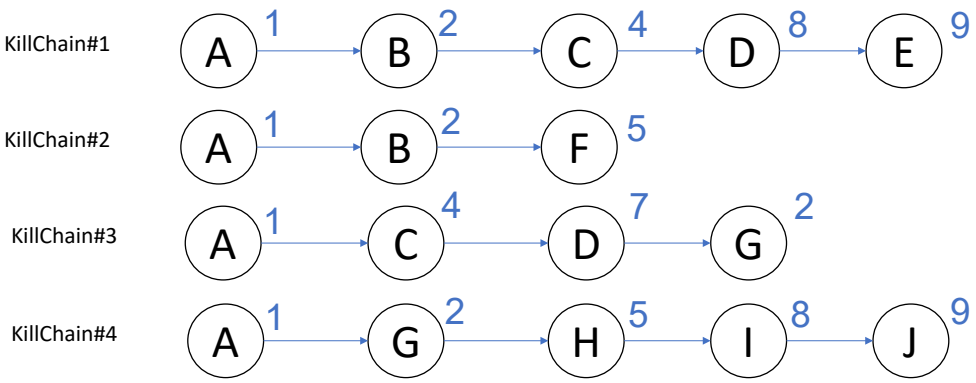
先ずは，起こりうる全キルチェーンを洗い出す（行 01）．このキルチェーンの集合を，KillChain$_{poss}$ と置く．次に，KillChain$_{poss}$ から，防御対象とするキルチェーンを抽出し，それらキルチェーンの集合を KillChain$_{def}$ とと置く（行 02）．KillChain$_{def}$ は，KillChain$_{def}$ ⊂ KillChain$_{poss}$ を満たす．次に，KillChain$_{def}$ 内で最長のキルチェーンを抽出し，キルチェーンの最大長$maxlength$を取得する（行 03）．次に，初期値を 1 とする整数型変数 N（行 04）と，初期値が空集合とするリスト型変数 checkedAttackStates（行 05）を宣言する．N は，防御層の層数を意味し，変数 checkedAttackStates は，優先順位が決定済みの攻撃状態を格納される．その後，初期値を 1 とする整数型変数 priority（行 06）を定義する．必要な変数定義後は，N = $maxlength$となるまで，繰り返し処理（while 文：行 07 - 行 19）を実行する．

最初の while 文は，防御層が未形成であるキルチェーン KillChain$_{undef}$ を抽出する処理である（行 08）．KillChain$_{undef}$ に関して，中には N-1 層目の防御層の優先順位を決め始めるタイミングで，すでに N-1 層以上の防御層を有するキルチェーンが存在している可能性がある．行番号 09 は，そのようなキルチェーンを KillChain$_{undef}$ から除外する処理である．

そして，二回目の while 文（行 10- 行 18）に到達するが，この処理ブロックを KillChain$_{undef}$ が空になるまで処理する．ブロック内の処理は，最初に，KillChain$_{undef}$ に最も含まれる攻撃状態を抽出し，リスト型変数 a$_{most}$ に格納する（行 11）．次に，a$_{most}$ と，priority を書き出す（行 12）．その後，攻撃状態の集合である a$_{most}$ の要素の数 Num(a$_{most}$)を計算し（行 13），priority に，Num(a$_{most}$)の和を新たな priority の値とする

（行 14）．そして，$KillChain_{undef}$のうち，a_{most}が経路に含まれるキルチェーンを$KillChain_{undef}$から除去する．a_{most}を経路に含まないキルチェーンから構成される集合を$KillChain_{nocont}$すると，次に，$KillChain_{nocont}$と$KillChain_{undef}$差集合を求め，その結果を$KillChain_{undef}$とする（行 15 - 行 16）．その後，a_{most}，および checkedAttackStates の和集合を求め，checkedAttackStates にその結果を格納する（行 17）．二回目の while 文完了後は，変数 N をインクリメントし，次層番の処理を開始する（行 19）．これらの処理を N = $maxlength$を満たすまで繰り返し実行する．

上記アルゴリズムの処理結果の例を Figure 5.10 に示す．



Point	Priority	Defense layer (DL)	The number of kill chains defensed
p_A	1	KillChain#1 (1st DL), KillChain#2 (1st DL), KillChain#3 (1st DL), KillChain#4 (1st DL)	4
p_B	2	KillChain#1 (2nd DL), KillChain#2 (2nd DL)	2
p_C	4	KillChain#1 (3rd DL), KillChain#3 (3rd DL)	2
p_D	7	KillChain#1 (4th DL), KillChain#3 (4th DL)	2
p_E	9	KillChain#1 (5th DL)	1
p_F	5	KillChain#2 (3rd DL)	1
p_G	2	KillChain#3 (2nd DL), KillChain#3 (2nd DL)	2
p_H	5	KillChain#4 (3rd DL)	1
p_I	8	KillChain#4 (4th DL)	1
p_J	9	KillChain#4 (5th DL)	1

Figure 5.10　An example about the output of the algorithm

この例の場合，$KillChain_{def}$ は，{ KillChain#1 , KillChain#2 , KillChain#3 , KillChain#4 }である．また，図中ノード右上の数字は，攻撃状態の優先順位である．各ポイントをp_A，p_B，p_C，… ，p_Jとすると，p_Aに防御層を置くことで，すべてのキルチェーンに対して第一層目の防御層を形成できることから，p_Aが最優先のポイントとなる．この場合，式(5.10)に示す集合Pは，$P = \{p_A\}$となる．

次は，p_Bとp_Cが，次に優先するポイント（同順）となり，次の優先順位はp_Cとなる．この要領で，$maxlength = 5$ まで処理を反復すると，Figure 5.10 に示す結果が得られる．

5.3.　評価

Figure 5.11 制御システムを例に，本提案方式を評価した．対象システムの構成を Figure 5.11 に示す．

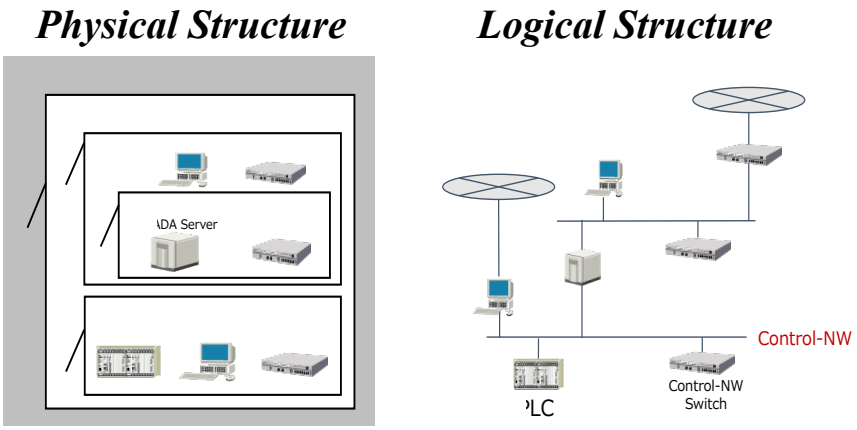


Figure 5.11　Physical and logical structure of the target system in this study

先ず，本システムのキルチェーンを洗い出す．これは，本提案手法によりモデル化したグラフから決定する．Figure 5.12 にそのグラフを表す．赤線のエッジは物理的な経路を意味し，黒線のエッジは，論理ネットワーク的な経路を意味する．

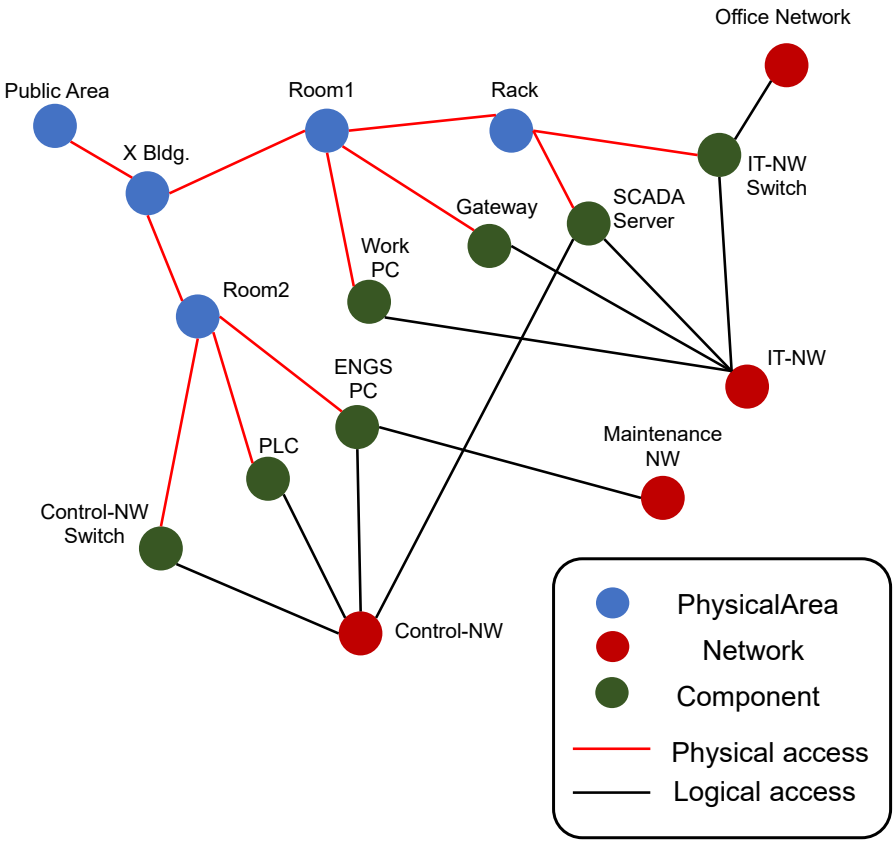


Figure 5.12　Graph structure of the target system in this study

次に，攻撃パスを求めるため，キルチェーンの始点と終点を定義する．今回の評価システムの場合，始点，終点の組み合わせ数は，$_{16}C_2$である．つまり，120 パターン存在することになる．これら組み合わせ

から事前分析の結果に基づき，始点と終点を絞り込む．今回，始点のルールとして，外部の物理的，または論理的なポイントとした．このルールに該当する始点は，Maintenance NW，Office Network，Public Area となる．次に，評価システムが担うプラント制御系アプリケーションの機能損失につながるノードを終点とした．Table 5.1 に，本評価における攻撃者の最終目標の一覧を示す．終点に該当するコンポーネントは SCADA Server，および PLC となる．結果的に，始点，終点の組み合わせパターンは全 6 パターン得られた．

Table 5.1　Final goals of an adversary

#	Category	Goal	The last attack
1	Unintended Control Actions	Manipulating controls of production line (Loss of safety)	• “#12 Change configurations and parameters” on “PLC” or “SCADA Server”
2	Stop Business Operations	Unintended stop control system (Loss of availability)	• “#10 Remote denial of service” on “PLC” or “SCADA Server” • “#11 Local denial of service” on “PLC”or “SCADA Server”

次に，Figure 5.77 のグラフを基にキルチェーンを決定する．例として，"SCADA Server"を最終目標とする攻撃パスを本グラフから計算すると 45 パターン得られ，"PLC"を最終目標とする攻撃パスは，73 パターンとなる．これらパターンから，キルチェーンを識別する．キルチェーンの識別は，Figure 5.77 のグラフにおけるノードとエッジに応じて対応する攻撃戦略と攻撃手法を割り当てる．今回，そのモデル ATT&CK，および Cyber kill chain model を参考に，キルチェーンモデルを設計した．設計したモデルを Figure 5.13 に示す．上記の共通モデルは，いずれも物理アクセスが含まれないため，物理アクセスのモデルを個別に追加した．

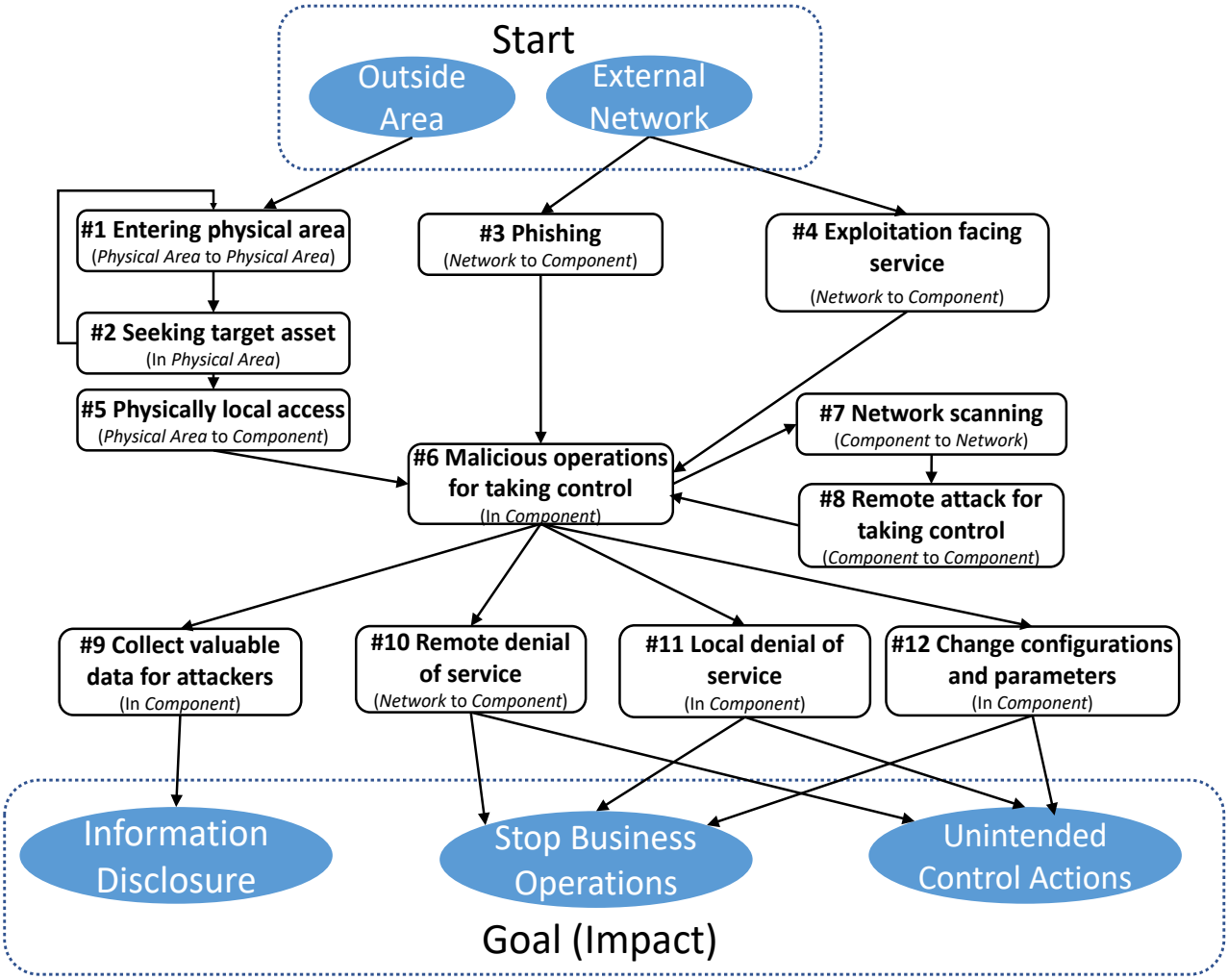


Figure 5.13　The kill chain model in this study

次のステップは，キルチェーン（攻撃シナリオ）を得るタスクである．前章に示した通り，グラフを網羅的に探索すると得られるキルチェーン数が膨大になり，その妥当性検証が困難となる．モデルを単純化し，得られるキルチェーンの数を制限するため，以下の前提を置く．

- 各ノードにおいて，攻撃が成功する可能性の度合いは同値である（前提 a）
- 論理アクセスに侵入した後は，物理的なアクセスには戻らない（前提 b）

上記の前提の元，今回想定する評価システムの場合，SCADA Server を最終目標とする攻撃パスは 9 パターン得られ，Figure 5.13 に示したキルチェーンに割り当てると，33 パターンのキルチェーンが得られた．また，PLC を最終目標とするケースの場合，攻撃パスは 9 パターン，そしてキルチェーンは 33 パターン得られ，両方を合わせると，66 パターン得られる．本評価システムと上記の前提で得られたキルチェーンの具体例を Figure 5.14 に表す．

Goal: Manipulating controls of production line

Kill chain #1

Step	Phase	From	To	Attack Tech
0	Initial Access	Office Network	Work PC	Phishing
1	Execution	Work PC	Work PC	Malicious operations for taking control
2	Scanning	Work PC	IT-NW	Network scanning
3	Lateral movement	Work PC	SCADA Server	Remote attack for taking control
4	Install and modify	SCADA Server	SCADA Server	Malicious operations for taking control
5	Impact	SCADA Server	SCADA Server	Change configurations and parameters

Kill chain #2

Step	Phase	From	To	Attack Tech
0	Physical Access	Public Area	X Bldg.	Entering physical area
1	Physical Access	X Bldg.	X Bldg.	Seeking target asset
2	Physical Access	X Bldg.	Room2	Entering physical area
3	Physical Access	Room2	Room2	Seeking target asset
4	Initial Access	Room2	ENGS PC	Physically local access
5	Execution	ENGS PC	ENGS PC	Malicious operations for taking control
6	Scanning	ENGS PC	Control-NW	Network scanning
7	Lateral movement	ENGS PC	PLC	Remote attack for taking control
8	Install and modify	PLC	PLC	Malicious operations for taking control
9	Impact	PLC	PLC	Change configurations and parameters

Figure 5.14　Examples of kill chains in this study

次に，獲得したキルチェーンに基づき，前述の優先順位付けアルゴリズムを適用した結果を示す．本評価システムにいて，最終的な優先順位を Table 5.2 に表す．Table 5.2 の結果は，優先度が上位の攻撃状態のみを抜粋した．Table 5.2 において，第一列は防御層の優先順位，第二列は攻撃位置，第三列は攻撃フェーズ，第四列は攻撃手法を示す．更に，第五列は防御層の番号，第六列は指定した点に対策を導入した場合に対処できるキルチェーン数である．

Table 5.2　Key points to be installed countermeasures

Priority	Point	Phase	AttackTech	Defense Layer	Killed Chains
1	SCADA Server	Install and modify	#6: Malicious operations for taking control	1st	51
2	PLC	Install and modify	#6: Malicious operations for taking control	1st	15
3	SCADA Server	Lateral Movement	#8: Remote attack for taking control	2nd	27
3	X Bldg.	Physical Access	#1: Entering physical area	2nd	
3	X Bldg.	Physical Access	#2: Seeking target asset	2nd	
6	ENGS PC	Execution	#6: Malicious operations for taking control	2nd	6
6	PLC	Lateral Movement	#8: Remote attack for taking control	2nd	
6	Control NW	Scanning	#7: Network scanning	2nd	
9	IT-NW Switch	Scanning	#6: Malicious operations for taking control	2nd	3
9	IT-NW	Scanning	#7: Network scanning	2nd	

結果として，SCADA Server への攻撃(#6)が，最も優先順位が高い結果となった．その次に優先度が高いのは，PLC に対する攻撃(#6)である．この結果は，最終目標である SCADA Server 及び PLC へ至るパスの内，最終目標に到達する直前にセキュリティ対策を実施することで，最も効果的に攻撃者の目標達成を阻止できることを意味する．

次に，防御層の優先度に関して触れる．SCADA Server，または PLC を終点とするキルチェーンの数は等しいものの，PLC を終点とするキルチェーンの中に，SCADA Server を経由するものが存在するため，結果的に，SCADA Server への攻撃に対処することで，51 パターンのキルチェーンに対処でき，PLC への攻撃に対処した場合よりも多くのキルチェーンに対策できる結果となった．

同順となっている SCADA Server に対するリモート攻撃(#8)，X Bldg.に対する物理侵入(#1)，および X Bldg.に対するターゲットの探索(#2)の三つの攻撃パターンに対して更に優先順位を付ける．これは，「攻撃者の侵入は可能な限りキルチェーンの早期段階で防ぐ」という防御側の心理に基づき，各キルチェーンの始点に近いポイントの優先度を高くする．その結果，同じ三番目の優先順位の攻撃パターンは，X Bldg.(#1)，X Bldg.(#2)，SCADA Server(#8)という優先順位となる．

以上により，セキュリティ対策について事前のモデル化が不要で対処すべき攻撃パターンの優先順位を機械的に得られることを確認した．

5.4.　考察

本提案方式は，オントロジに基づき構造的にモデル化された制御システムモデルに対し，サイバーキルチェーンモデル，および多層防御モデルのキルチェーンのみから優先的なセキュリティ対策を決定する手法である．本手法は，対処すべき複数キルチェーンの攻撃状態の重複度合いを評価し，最も重複する攻撃状態を優先的な対策実施ポイントとして決定し，当対策実施ポイントに対応するセキュリティ対策を同定することで，少ない対策で多くの脅威に対処できる効果的なセキュリティ対策を得る手法である．

先行研究との明確な差異は，事前に利用可能なセキュリティ対策を特定し，コスト情報を始め詳細な対策モデルの事前定義が不要である点である．効果的なセキュリティ対策を得る効果という文脈では，先行研究の手法は，単一の脅威シナリオに対して，最適な対策の獲得を試みるが，本方式は，網羅的に識別された多数の脅威シナリオを基に，全シナリオに対応可能な対策を得る点が際である．例えば N. Poolsappasit らの手法[135]は，脅威シナリオを攻撃木から成るベイジアンネットワークとして構成し，ベイジアンネットワークを基に，最も損失（リスク）が低くなるように各ノード（攻撃パターン）に対して対策パターンを講じるものである．これは，単一のベイジアンネットワークから得られるものであり，複数の脅威シナリオを基に得るためには，脅威シナリオ毎でベイジアンネットワークを構成し，最適なセキュリティ対策を同定し，各脅威シナリオで得られたセキュリティ対策のパターンを統合する必要がある．通常，対処すべき脅威シナリオのパターンは一つのシステムに対して複数存在するため，既存研究の手法は非効率である．したがって，本提案手法は，膨大な複数の脅威シナリオに対して，効果的なセキュリティ対策を得ることができる．反面，脅威シナリオの事前の網羅が本手法では必須となるが，これはリスク識別に示したモデルにより，網羅的に脅威シナリオを得ることが可能となる．

本手法は，優先順位が高い攻撃に絞り，その攻撃に効果的に対処すべきセキュリティ対策を決定することで，推奨されるセキュリティ対策を得られる．一方で，各攻撃パターンに対して効果的に対処できるセキュリティ対策の実体を事前知識が無い状態で決定することは困難である．そのため，セキュリティ対策の実体を得るためには，知識に基づくセキュリティ対策実体の情報モデル（セキュリティ対策知識モデル）を構築する必要がある．セキュリティ対策知識モデルのイメージを Figure 5.15 に示す．このモデルでは，各攻撃パターンに対して効果があるセキュリティ対策の実体を知識に基づき定義する．各攻撃状態に対して効果のあるセキュリティ対策を得るためには，本モデルを基に該当する対策を出力する．セキュリティ対策知識モデルを一度構築できれば，それを再利用することも可能である．

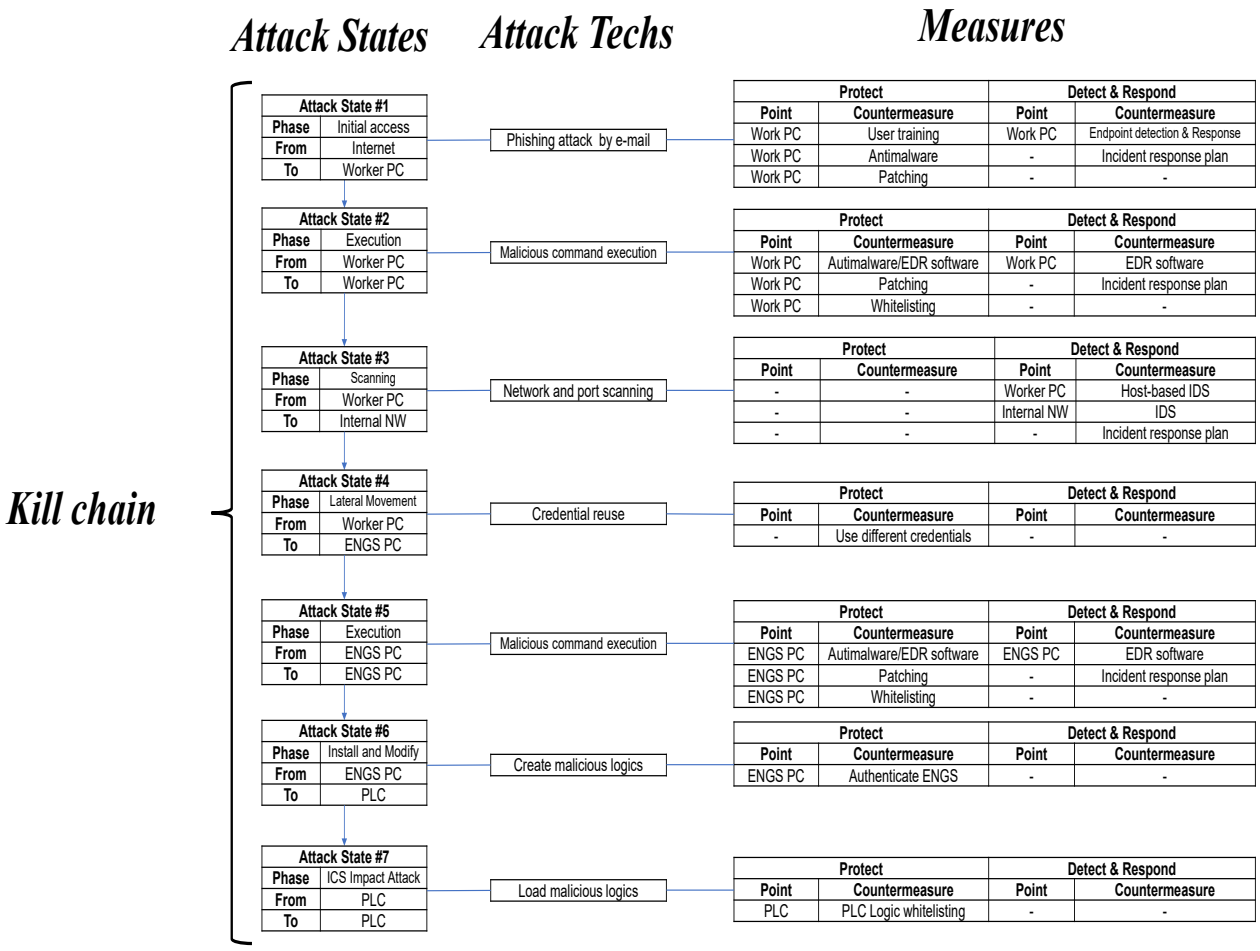


Figure 5.15　Image of the security measure knowledge model

セキュリティ対策の実体を獲得するには，セキュリティ対策知識モデルを構築する必要があり，そこには依然セキュリティ対策に関する事前知識が必要となる．この事前知識は，専門家によって構築されるが，セキュリティ対策知識モデルに活用可能な事前知識の例として，MITRE ATT&CK Mitigations があり，これらを活用することで，セキュリティ対策実体を得る上で属人性を排除できる．

最後に，リスク対処過程におけるプロセス効率化の効果について述べる．リスク対処過程は，リスク評価過程において決定された優先的な脅威シナリオを基に，それらに対処する効果的なセキュリティ対策を決定する．この作業を人手で実施する場合，優先的な脅威シナリオそれぞれに対して，効果のある対策を試行錯誤的に決定する必要があり，そのプロセスは属人的である．Table 5.3 に本提案手法により，属人性の排除，または，自動化を達成した項目を示す．本提案手法により，取り組むべき脅威の識別，攻撃パターンの優先順位付けの自動化を達成した．また，効果のあるセキュリティ対策パターンを得るためには，各攻撃パターンに対し，効果のあるセキュリティ対策を知識に基づき関連付け，セキュリティ対策知識モデルを構築する必要があるが，一度関連付けた後はそのパターンを再利用することが可能である．ただし，このセキュリティ対策知識モデルは，定期的にアップデートする必要がある．

Table 5.3 Achievement status of elimination of individuality or automation by the proposed method

#	Process	Description	Achievement
1	**Identify threat scenarios**	Identify threat scenarios to be addressed	Complete
2	**Prioritize critical attack patterns**	Prioritize attack patters from the identified threat scenarios	Complete
3	**Identify effective measures**	Identify effective cybersecurity measures critical for points	Partial
4	**Cost estimation**	Estimate the total cost for cybersecurity measures identified	None
5	**Finalize requirements**	Finalize cybersecurity measures requirements for a target system	None

一方で，セキュリティ対策に係る費用の絶対値を，属人性を排除しつつ獲得し，最終的に効果的なセキュリティ対策仕様を機械的に獲得することは本提案手法では未達成である．多くの先行研究が注目するセキュリティ対策コストの把握は，制御システムオーナにとって最も興味がある事柄であることは明らかである．本提案手法は，利用できるセキュリティ対策が未定であっても，対策の優先順位付けが可能である点で有用であるが，セキュリティ対策が未定の故，コストの見積もりができない．コスト効果を見積もるためには，やはり候補となるセキュリティ対策の知識を活用が必須である．本提案手法は対策コストに基づくアプローチと統合できる可能性がある．例えば，最初にキルチェーンのみから，対策すべき攻撃パターンの優先順位を決めた後に，優先度が高い攻撃パターンに対して効果的なセキュリティ対策を割り当てる際に，候補となる対策のコスト情報が既知である場合，セキュリティ対策に係るコスト見積もりが可能となる．対策コストモデルのオントロジについては，引き続き検討の余地がある．セキュリティ対策の表現モデルについて，報告書らによってセキュリティ対策の分類軸を議論[138]しており，そのモデルに基づくセキュリティ対策の効果とコストの詳細モデル設計を引き続き検討する．

5.5. 本章のまとめ

制御システムのセキュリティ対策計画の立案を自動化する技術に関し，多くの先行研究でセキュリティ対策の論理モデルを構築し，低コストかつ効果的なセキュリティ対策を得る手法が提案されている．これら先行研究の手法は，対策の実践コストやリスクに対する影響の詳細モデルが既知である前提の基，機能するものであり，特にセキュリティ対策が未定の場合は利用できない課題があった．本論では，キルチェーンモデルで表現された脅威シナリオのみを用いて，多層防御モデルに基づき，優先して対策すべき攻撃パターンを機械的に決定し，セキュリティ対策仕様の自動化に獲得する手法を提案した．この手法は，セキュリティ対策が不定な状況であっても，最大の効果が得られるセキュリティ対策を得ることができる．

一方，攻撃パターンから最終的にセキュリティ対策を決定するには，コストおよび効果に関するセキュリティ対策のモデルが必須であり，本論の提案手法とあわせて，オントロジに基づく対策コストのモデル化も今後の研究対象である．

第6章 結論

6.1. オントロジ駆動型モデリングのセキュリティエンジニアリングへの効果

制御システムオーナが，脅威モデリングを実施するにあたり，脅威モデリングに関する知識の再利用性向上や自動化により効率化し，そのプロセスの属人性排除と，設計コストの低減を支援するシステムを実現するという本論の大目的に対して，脅威モデリングに対するオントロジ駆動型モデリングの有用性を評価した．

既述の通り，脅威モデリングは大きく，セキュリティ脅威を識別するプロセス（リスク識別），識別した脅威を評価し，対処すべき脅威を選定するプロセス（リスク評価），対処すべき脅威を基に，対処策を立案するプロセス（リスク対処）から成る．

第 3 章に示したリスク識別過程における既存手法は，網羅的な脅威シナリオの自動抽出の実現にあたり，抽出された脅威シナリオの優先順位付けの観点で課題があった．リスク識別過程において，本章で着眼した課題は，脅威シナリオモデル化手法として，モデルの再利用性を担保する脅威アクターの状態モデルに基づく脅威シナリオの表現モデルの確立である．本章の提案手法は，リスク識別過程において，脅威シナリオをモデル化するものであり，

既知の脅威アクターの制御システム上の挙動をオントロジに基づき表現する．脅威シナリオのモデル化にあたり，基本モデルとして，Diamond model に基づき，脅威アクターの挙動を，状態遷移モデルを用いて表現し，始点から一連の状態履歴を攻撃シナリオとして構造的に表現する．この表現手法により，システム要素に応じた攻撃シナリオ要素のパターン化でき，既知の脅威アクターに応じた制御システムの構成に応じ，攻撃シナリオを再現できる．また，モデル化された脅威アクターに対し，最終目標に到達可能な脅威シナリオのみを抽出できることから，特に優先的な脅威シナリオに限定して識別できる．

一方で，未知の攻撃者の挙動を再現できない点や，制御システムの規模が増大するほど計算量が増加する点，モデル化誤差の観点で課題が残る．特に，計算量の議論では，本論で実装したアルゴリズムでは，制御システム内の資産数をn，$m \in \mathbb{N}$とした場合，脅威シナリオを識別する計算量が$O(2^n)$から$O(n^m)$の範囲となり，規模が多いシステムの場合は，計算爆発が起こる恐れがある．

第 4 章に示したリスク評価過程における既存手法は，攻撃対象の脆弱性情報のみを対象としており，攻撃者自身の要素が考慮されていない点で課題があった．リスク評価過程において，本章で着眼した課題は，属人性を排除しつつセキュリティリスクの見積もる手段として，前述の脅威シナリオ表現モデルに基づくリスク計算モデルを提案することである．

本章の提案手法は，第 3 章に示した脅威シナリオの表現モデルに対して，攻撃が成立する可能性を定量的に表現する手法である．この手法は，脅威シナリオにおける侵入起点から最終目標に至るまでの確率モデルを表現できる．このモデルは，同程度の悪影響につながる攻撃シナリオに関して，従来の定性的な手法と比べて，詳細な優先順位付けを可能とする．更に第 3 章に示した脅威シナリオの表現モデルを用いていることから，リスク評価モデルの再利用性も担保される．

残課題として，情報の蓄積が不十分な攻撃者の発現確率や，特定攻撃手法の成功確率を実験的に獲得する必要がある点がある．加えて，影響度の定量化も検討する必要がある．これは，制御システムの稼働停止や意図しない動作時の損害を評価する必要があり，この評価は事業影響度分析など自動化困難なプロ

セスを実施する必要である．

第 5 章に示したリスク対処過程における既存手法は，コスト情報といった利用可能なセキュリティ対策モデルが既知である必要があり，それを事前に得ることができない場合に手法を適用できない点が課題であった．本章で着眼した課題は，高リスクの脅威シナリオに対し，セキュリティ対策の前提知識不要で優先的なセキュリティ対策を実践すべき箇所を機械的に決定する手法を提案することである．本章の提案手法は，対策立案過程において，利用可能なセキュリティ対策が不明な状況において，優先すべきセキュリティ対策の実施箇所を機械的に決定できることを示した．本手法により優先すべき箇所における攻撃パターンと，当該攻撃パターンに対応するセキュリティ対策を割り出すことで，自ずと優先すべき対策を機械的に決定できる．

一方で，完全なコスト対効果を示すには，セキュリティ対策の導入，運用に係るコストの定量化は避けられない．現実として，実際のセキュリティ対策は，実際の対策毎で脅威に対する効果とコストが異なる．例えば，対策として暗号化を一つとっても，そのアルゴリズムや鍵管理方式は多様であり，各方式でコストと脅威への対策効果は異なる．セキュリティ対策毎のコストと脅威に対する効果のモデル化が必要であるが，そのモデルは事例を元に決定せざるを得えない．したがって，セキュリティ対策のコストと脅威に対する効果についての知識の蓄積と再利用性の確保は，継続して検討すべき課題である．

以上の議論を纏めた結果を Table 6.1 に示す．脅威モデリング各ステージに対して，オントロジ駆動型モデリング基づく，各提案手法を適用した結果，残課題は残るものの，命題とした課題を解決できることを示した．オントロジ駆動型モデリングの産業面に対する効果として，制御システムの脅威モデリングの省人化に貢献し，セキュリティエンジニアリングプロセスの設計コストを低減できる．

Table 6.1　Summary of results of the ontology-driven modeling proposed in this study

Stage	Summary of Method	Achievement	Remaining challenges
Risk identification	Representing behaviors of threat actors using a state transition model based on the Diamond model	- Patterning attack scenario elements according to system elements - Reproducing attack scenarios by known threat actors on customized ICS - By identifying only threat scenarios that reach final goals, prioritized threat scenarios can be obtained eliminating individual dependencies	- Reproducing the behavior of unknown attackers - There is a possibility of a computation explosion for large scale ICS,
Risk evaluation	Quantitatively expressing the likelihood of attacks based on the proposed threat actor behavior model	- Expressing the probability from the starting point of the intrusion to the final target in a threat scenario - Detailed prioritization compared to traditional qualitative methods can be obtained automatically - Ensuring reusability of the probability models by utilizing the structured threat representation models	- The probability of occurrence of an attacker with insufficient information accumulation and the numerical values of specific parameters should be determined based on actual measured data - Quantification of impact severity
Risk handling	Determining priorities of where to implement security measures based on cyber kill chains and defense-in-depth models	- Determining where to implement security measures where the available security measures are unknown - Determining effective security measures for the prioritized points automatically	- Quantification of implementing cost and operating cost - Planning a holistically optimized security measures for a limited budget - Modeling effectiveness and cost against threats

最後に本論で提案する手法の属人性排除，自動化の効果について Table 6.2 に示す．制御システムオーナが，脅威モデリングを実施するにあたり，脅威モデリングに関する知識の再利用性向上や自動化により効率化し，そのプロセスの属人性排除と，設計コストの低減を支援するシステムを実現するという本論の命題に対し，脅威モデリングの各過程において，モデル定義とその再利用により，そのプロセスの属人性排除，自動化を部分的に達成できることを示した．一方で，脅威モデリングの完全自動化には依然課題が残るため，引き続き研究の余地がある．

Table 6.2　Achievement status of elimination of individuality or automation in this dissertation

	#	Process	Description	Achievement
Risk identification	**1**	**Identify threat actors**	Identify assumed threat actors	Completed
	2	**Identify impactful events**	Identify impactful events for the system owner of the target system	None
	3	**Identify of attack scenarios**	Obtain a comprehensive set of attack scenarios for the target system	Completed
Risk evaluation	**1**	**Estimate degree of impact**	Estimate the degree of impact caused by a cyber attack	None
	2	**Estimate degree of likelihood**	Estimate the degree of likelihood which the threat is likely to materialize	Partial
Risk handling	**1**	**Identify threat scenarios**	Identify threat scenarios to be addressed	Complete
	2	**Prioritize critical attack patterns**	Prioritize attack patters from the identified threat scenarios	Complete
	3	**Identify effective measures**	Identify effective cybersecurity measures critical for points	Partial
	4	**Cost estimation**	Estimate the total cost for cybersecurity measures identified	None
	5	**Finalize requirements**	Finalize cybersecurity measures requirements for a target system	None

本論において提案する手法の本質は，オントロジ駆動に基づき制御システムセキュリティに係る要素のモデルを構築し，そのモデルを再利用することで，プロセスの属人性排除と自動化を達成する．すなわち，本手法によって得られるモデルの妥当性についても議論が必要である．モデルの妥当性は，実際の脅威事例や対策事例と比較することで，矛盾有無を確認することで検証した．理想的には，形式検証などの数学的な手法に基づき，モデルの妥当性を示すことである．モデルの妥当性を検証する条件についても不明瞭な点が多い．そのため，モデル妥当性の評価も属人性が残ることが実情である．モデルの妥当性については，産業分野では認証機関による第三者認証によって実現されているケースが多い．そのため，制御システムセキュリティに係る全ての事象を形式的に表現できれば，本質的に人間の属人化された知識が不要で科学的にモデルを表現できることが理想であるが，現状，制御システムセキュリティに係るモデルの理論は依然発展途上である．そのため，現在の産業分野では，人間の知識に依存して妥当性が決められることが大半である．モデルの妥当性は，制御システムセキュリティの詳細モデルが明らかになるにつれ，より厳密に評価できるようになる可能性がある．

また，制御システムセキュリティのモデルが具体的表現される過程においては，抽象モデルの合意形成の仕組みは必須である．実際に ATT&CK や CWE といったモデルは，一部の専門化団体によって議論さ

れ，開発された標準的なモデルである．これら標準モデルは事例に基づきモデル化されたものであり，科学的に証明された第一原理に基づくモデルでは無いもの，これらモデルが受け入れられているのは，オープンコミュニティにおいて合意形成の元で開発されたモデルであることが理由である．すなわち，産業分野への応用という文脈において，オープンコミュニティにおいて合意形成を促すことで，合意形成の元，妥当であると判断されたモデル（参照モデル）を構築し，確立した参照モデルを活用することで，モデルの属人性の排除を達成することも一つの手段である．

6.2. ベンチマーク

本論で提唱するオントロジ駆動型モデリングに基づく制御システム向け脅威モデリング手法に関連する先行研究とのベンチマークを議論する．本論の提案手法は，脅威モデリング各プロセス（リスク識別，リスク評価，リスク対処）を対象とする．

提案手法のベンチマークを既存研究との比較を Table 6.32 に示す．これら先行研究は，脅威モデリングの自動化を目的としたものである．#2~#8 は，リスク識別と評価に関する手法であり，#9~#12 はリスク対処に関する手法である．リスク識別と評価に関する手法に関して，それぞれ手法は異なるものの，脅威事象を論理的にモデル表現し，合理的な脅威シナリオを導き出すことを目指している．これらの方法の共通の特徴は，脅威または攻撃シナリオを，再利用性可能な形でモデル化することで，脅威イベントの再利用性を確保する．また，方法毎で攻撃対象とする制御システムをモデル化する範囲は異なる．

Table 6.3 Benchmark for this study

#	Work	Objective	Reusability	Express detailed system structure	Define capabilities of adversary	Evaluate dynamic actions	Evaluate likelihood	No Predefined measures	Multi-Threat Scenarios	No Quantify effect of measures	Consider budget for measures
1	This work	Determine priority scenarios and critical points to secure	✓	✓	✓	✓	✓	✓	✓	✓	-
2	G. Falco, et al.[53]	Comprehensive threat identification	✓	-	-	-	-	N/A	N/A	N/A	N/A
3	G. Chu, et al.[131]	Attack simulation	✓	-	-	✓	-	N/A	N/A	N/A	N/A
4	P. Johnson, et al.[132]	Describe threat event	✓	✓	-	-	-	N/A	N/A	N/A	N/A
5	R. Khan, et al.[133]	Comprehensive threat identification	✓	-	-	-	-	N/A	N/A	N/A	N/A
6	P. Johnson, et al.[134]	Automatic threat identification	✓	✓	-	-	-	N/A	N/A	N/A	N/A
7	A. Ekelhart, et al.[135]	Attack simulation	✓	✓	-	-	-	N/A	N/A	N/A	N/A
8	M. Mohsin, et al.[108]	Identify likelihood of attack scenarios	✓	✓	✓	-	✓	N/A	N/A	N/A	N/A
9	W. Widel, et al. [60]	Simulate attacker's time to compromise	N/A	N/A	N/A	N/A	N/A	-	-	-	✓
10	O. Stan, et al. [61]	Cost-aware measures heuristic search	N/A	N/A	N/A	N/A	N/A	-	-	-	✓
11	L. Wang, et al. [62]	Assign metric values for attack graphs and measures	N/A	N/A	N/A	N/A	N/A	-	✓	-	✓
12	Y. Fei, et al. [63]	Assign quantitative security attribute information	N/A	N/A	N/A	N/A	N/A	-	-	-	✓

本論の提案手法は，制御システムの内部構造をオントロジに基づきモデル化し，サイバー攻撃に関する様々な属性を持つグラフモデルとして表現し，攻撃を試みる脅威アクターの能力に基づき攻撃の成功または失敗を決定する．これに類するアプローチとして，M. Mohsin らの提案方法(#8)がある．彼らの手法は，脅威アクターの能力を確率的に表現し，定量的なリスク表現を提供するものである．彼らの方法が離散的で独立した攻撃活動に対するモデル化に重点を置いていることに対し，本提案手法は，数理モデル

として構造化表現された制御システムに対し，各攻撃によって引き起こされる攻撃状態の変化を取り入れることで，脅威アクター攻撃過程における動的な特性を反映する．本提案手法は，この特性に基づき，脅威シナリオを表現，特定し，そのリスクを評価することで，実際の脅威アクターの行動を構造的に表現する．その結果，属人性を排除しつつ，詳細に表現された脅威シナリオの優先順位付けを可能にする点において優位である．

リスク対処の観点の差異は，すでに第 4 章において示した通り，セキュリティ対策効果の詳細モデル化を必須とする W. Wideł ら等の先行研究の手法と比べ，本提案手法は，セキュリティ対策のモデルが不定であっても利用できる点である．更に既存研究の提案は，脅威シナリオ単位での評価を前提とするが，本提案は，網羅的に得た多数の脅威シナリオに対し，すべてのシナリオに有効な対策を決定できる点も特徴である．その反面，本提案は，予め網羅的に脅威シナリオを識別することが必須である．

その一方，先行研究が注目するセキュリティ対策に係るコストを把握することは，制御システムオーナにとって重要な事柄である．本手法は，セキュリティ対策が未定であっても適用できる手法であるが，対策ポイントと実際の対策を紐づける際に，コストを含む既知のセキュリティ対策モデルの知識を関連付けると，システム全体の対策実践コストの把握も可能となることが期待できる．

6.3. 今後の展望

最後にオントロジ駆動型モデリングによる制御システムのセキュリティエンジニアリングについて，今後の展望を述べる．オントロジ駆動型モデリングは，対象のオントロジを明らかにし，公理や知識，実測的な結果に基づき，対象の要素と内部構造を数理的なモデルで表現することが本質ある．本論では，制御システムの脅威モデリングにおけるリスク識別，リスク評価，およびリスク対処の三過程について，オントロジ駆動型モデリングに基づく提案手法と，その効果と残課題を示した．今後は各残課題に対する解決策を研究し，脅威モデリングの完全自動化を目標に進めたい．

セキュリティエンジニアリング全体において，リスク対処後は，制御システムに対する対策の実装，セキュリティ運用，セキュリティ検証・監査といったプロセスが存在するが，これらプロセスも金銭的，人的コストが発生する．脅威モデリング後のプロセスにおいても，オントロジ駆動型モデリングを適用することで，各プロセスの効率化を期待できる．例えば，対策の実装においては，IaC (Infrastructure as Code)[145]の活用によるセキュリティ機能の自動適用が挙げられる．IaC の適用が進むクラウドベースのシステムの場合は比較的容易に実現できるが，多様なハードウェアから成る制御システムは現状容易でない．この対策として，AAS (Asset Administration Shell)[145][146]などのデジタルツイン等の技術を介してセキュリティ機能を自動適用するといった方法が考えられる．IaC や AAS は，モデル化対象とするシステム構成要素のオントロジを明らかにし，それらをモデルとして再現することから，オントロジ駆動型モデリングの代表例といえる．一方で，稼働中の制御システムや，既設の制御システムに対するセキュリティ対策の自動適用は依然確立した技術が無く，研究の余地がある．また，コントローラ上で動作する制御プログラムに対し，セキュリティ対策を実装し，自動的に対策を適用とする取り組みも研究報告もある[51]．システム側だけでなく，制御論理そのものに対するセキュリティ対策の自動適用も含め，検討に含める必要がある．

セキュリティ運用や，セキュリティ検証・監査についても，各プロセスのオントロジを明らかにし，そ

のオントロジに基づくモデル化により，各プロセスの遂行に関する知識の再利用性の確保し，プロセスの効率化や自動化を期待できる．一方で，プロセスの実施形態は制御システムにより様々であることから，その実現は多くの技術課題が存在する．オントロジ駆動型モデリングをセキュリティ運用や監査といったプロセスに適用する場合の課題と，その解決手段についても，今後の研究対象といえる．

纏めると，オントロジ駆動型モデリングはセキュリティエンジニアリングの全プロセスに対して適用できる可能性がある．今後はオントロジ駆動型モデリングによる脅威モデリングの完全自動化の実現だけでなく，セキュリティエンジニアリングにおける他のプロセスに対してもオントロジ駆動型モデリングのフィージビリティスタディの実施と，各プロセスに適したオントロジ駆動型モデリングに基づく手法を提案していきたい．そして，セキュリティエンジニアリングプロセスが完全自動化でき，あらゆる制御システムに対するセキュリティ対策の完全実施された世界を実現し，安全かつセキュアな社会インフラを実現に貢献したい．

参考文献

[1] N. Sands and I. Verhappen, "A Guide to the Automation Body of Knowledge, 3rd Edition," International Society of Automation (ISA), Mar. 2018.

[2] M. Hudedmani, R. Umayal, S. K, Kabberalli and R. Hittalamani, "Programmable logic controller (PLC) in automation" *Advanced Journal of Graduate Research*, 2(1), pp.37-45, Jul. 2017.

[3] "Distributed control system," https://en.wikipedia.org/wiki/Distributed_control_system, Accessed on 24th Apr. 2024.

[4] "SCADA," https://en.wikipedia.org/wiki/SCADA, Accessed on 24th Apr. 2024.

[5] ANSI/ISA-95.00.03-2005, "Enterprise-Control System Integration, Part 3: Models of Manufacturing Operations Management," International Society of Automation (ISA), Jul. 2005.

[6] H. Xu, W. Yu, D. Griffith and N. Golmie "A survey on industrial Internet of Things: A cyber-physical systems perspective," *IEEE Access*, Vol.6, 78238-78259, Dec. 2018.

[7] IEC TS62443-1-1, "Industrial communication networks – Network and system security - Part 1-1: Terminology, concepts and models," IEC, Jul. 2009.

[8] W. Theodore, "The Purdue enterprise reference architecture," *Computers in industry*, Vol.24.2-3: pp.141-158, Sep. 1994.

[9] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule and M. Thompson, "Guide to Operational Technology (OT) Security," National Institute of Standards and Technology (NIST), Sep. 2023.

[10] Trend Micro, "CONFICKER," https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware /conficker, Accessed on 24th Apr. 2024.

[11] Trend Micro, "RAMNIT," https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware /ramnit, Accessed on 24th Apr. 2024.

[12] D. Kushner, "The real story of stuxnet," *IEEE Spectrum*, Vol.50.3: pp.48-53, Feb. 2013.

[13] "Havex," https://en.wikipedia.org/wiki/Havex, Accessed on 24th Apr. 2024.

[14] "Shamoon," https://en.wikipedia.org/wiki/Shamoon, Accessed on 24th Apr. 2024.

[15] J. Nazario,"BlackEnergy DDoS Bot Analysis," Arbor Networks. Archived from the original on 21 Feb. 2020, Accessed on 24th Apr. 2024.

[16] CISA, "CrashOverride Malware," ICS-CIRT Alert, https://www.cisa.gov/news-events/alerts/2017/06/12/crashoverride-malware, Accessed on 24th Apr. 2024.

[17] G. Samuel, "Triton: hackers take out safety systems in watershed attack on energy plant," The Guardian, Oct. 2019.

[18] Symantec Threat Hunter Team, "Dragonfly: Western energy sector targeted by sophisticated attack group," https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/dragonfly-energy-sector-cyber-attacks, Accessed on 24th Apr. 2024.

[19] J. O'Leary, J. Kimble, K. Vanderlee and N. Fraser, "Insights into Iranian Cyber Espionage: APT33 Targets Aerospace and Energy Sectors and has Ties to Destructive Malware," Mandiant, https://www.mandiant.com/

resources/blog/ apt33-insights-into-iranian-cyber-espionage, Accessed on 24th Apr. 2024.

[20] B. Bracken," ICS Ransomware Danger Rages Despite Fewer Attacks," Dark Reading, https://www. darkreading.com/ics-ot-security/ics-ransomware-rages-fewer-attacks, Accessed on 24th Apr. 2024.

[21] E. Nakashima, "Foreign hackers targeted U.S. water plant in apparent malicious cyber attack, expert says," https://www.washingtonpost.com/blogs/checkpoint-washington/post/foreign-hackers-broke-into-illinois-water-plant-control-system-industry-expert-says/2011/11/18/gIQAgmTZYN_blog.html, Accessed on 24th Apr. 2024.

[22] "Flame (malware)," https://en.wikipedia.org/wiki/Flame_(malware), Accessed on 24th Apr. 2024.

[23] BSI, "The State of IT Security in Germany 2014," https://www.bsi.bund.de/SharedDocs/ Downloads/EN/ BSI/Publications/Securitysituation/IT-Security-Situation-in-Germany-2014.pdf, Accessed on 24th Apr. 2024.

[24] R. Spenneberg, "PLC-Blaster: A Worm Living Solely in the PLC," *Black Hat Asia 16*, 1-16, Mar, 2016.

[25] "WannaCry ransomware attack," https://en.wikipedia.org/wiki/WannaCry_ransomware_attack, Accessed on 24th Apr. 2024.

[26] "VPNFilter," https://en.wikipedia.org/wiki/VPNFilter, Accessed on 24th Apr. 2024.

[27] K. Higgins, "Ryuk Ransomware Hit Multiple Oil & Gas Facilities, ICS Security Expert Says," https://www.darkreading.com/threat-intelligence/ryuk-ransomware-hit-multiple-oil-gas-facilities-ics-security-expert-says, Accessed on 24th Apr. 2024.

[28] J. Nirmal, "Breach at Kudankulam nuclear plant may have gone undetected for over six months: Group-IB," https://economictimes.indiatimes.com/news/politics-and-nation/breach-at-kudankulam-nu clear-plant-may-have-gone-undetected-for-over-six-months-group-ib/articleshow/79412969.cms, Accessed on 24th Apr. 2024.

[29] Trend Micro, "What You Need to Know About the LockerGoga Ransomware," https://www. trendmicro.com/vinfo/us/security/news/cyber-attacks/what-you-need-to-know-about-the-lockergoga-ransomware, Accessed on 24th Apr. 2024.

[30] B. Hunter and F. Gutierrez, "EKANS Ransomware: A Malware Targeting OT ICS Systems," Fortiguard Labs Threat Research, https://www.fortinet.com/blog/threat-research/ekans-ransomware -targeting-ot-ics-systems, Accessed on 24th Apr. 2024.

[31] J. Bergal, "Florida Hack Exposes Danger to Water Systems," https://stateline.org/ 2021/03/10/florida-hack-exposes-danger-to-water-systems/, Accessed on 24th Apr. 2024.

[32] J. Easterly, "The Attack on Colonial Pipeline: What We've Learned & What We've Done Over the Past Two Years," https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years, Accessed on 24th Apr. 2024.

[33] ESET Research, "Industroyer2: Industroyer reloaded," https://www.welivesecurity.com/ 2022/04/12/industroyer2-industroyer-reloaded/, Accessed on 24th Apr. 2024.

[34] CISA, "Cyber-Attack Against Ukrainian Critical Infrastructure," ICS-CIRT Alert, https://www. cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01, Accessed on 24th Apr. 2024.

[35] W. Knowles, D. Prince, D. Hutchison, J. Disso and K. Jones, "A survey of cyber security management in industrial control systems," *International Journal of Critical Infrastructure Protection*, Vol.9, pp.52-80, Jun. 2015.

[36] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby and K. Stoddart, "A review of cyber security risk assessment methods for SCADA systems," *Computers & Security*, Vol. 56, pp. 1-27, Feb. 2016.

[37] Q. Qassim, N. Jamil, M. Daud, A. Patel and N. Ja'affar, "A review of security assessment methodologies in industrial control systems," *Information & Computer Security*, Vol. 27, No.1, 47-61, Feb. 2019.

[38] E. Byres, F. Matthew and M. Darrin, "The use of attack trees in assessing vulnerabilities in SCADA systems," in *Proceedings of the international infrastructure survivability workshop (IISW'04)*, May 2004.

[39] D. Zhang, Q. Wang, G. Feng, Y. Shi and A. Vasilakos, "A survey on attack detection, estimation and control of industrial cyber–physical systems," *ISA Transactions*, Vol.116, pp.1-16, Oct. 2021.

[40] Y. Hu, A. Yang, H. Li, Y. Sun and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, Vol. 14, No. 8, Aug. 2018.

[41] B. Zhu and S. Sastry, "SCADA-specific intrusion detection/prevention systems: a survey and taxonomy," in *Proceedings of the 1st workshop on secure control systems (SCS)*, Vol. 11, p.7-23, Apr. 2010.

[42] E. Yılmaz and S. Gönen, "Attack detection/prevention system against cyber attack in industrial control systems," *Computers & Security* Vol. 77, pp. 94-105, Aug. 2018.

[43] W. Yang and Q. Zhao, "Cyber security issues of critical components for industrial control system," in proceedings of *2014 IEEE Chinese Guidance, Navigation and Control Conference*, pp. 2698-2703, Aug. 2014.

[44] S. Nazir, S. Patel and D. Patel, "Assessing and augmenting SCADA cyber security: A survey of techniques," *Computers & Security*, Vol. 70, pp. 436-454, Sep. 2017.

[45] T. Morris and W. Gao, "Industrial control system cyber attacks," in Proceedings of *the 1st International Symposium on ICS and SCADA Cyber Security Research*, pp. 22-29. Sep. 2013.

[46] M. Alanazi, A. Mahmood and M. Chowdhury "SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues," *Computers & Security*, Vol. 125, 103028, Feb. 2023.

[47] D. Ding, Q. L. Han, Y.Xiang, X. Ge and X. M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems, " *Neurocomputing*, Vol. 275, No. 38, pp. 1674-1683, Jan. 2018.

[48] T. Ikezaki, O. Kaneko, K. Sawada and J. Fujita, "Poisoning attack on VIMT and its adverse effect," *Artificial Life and Robotics*," Vol. 13, pp.1-9, Nov. 2023.

[49] D. Ding, Q. L. Han, Z. Wang and X. Ge, "A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems," in *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 5, pp. 2483-2499, May 2019.

[50] A. Kalam, "Securing SCADA and critical industrial systems: From needs to security mechanisms," *International Journal of Critical Infrastructure Protection*, Vol. 32, 100394, Mar. 2021.

[51] K. Sawada, "Model-based cybersecurity for control systems: Modeling, design and control," Proceedings of *2017 56th Annual Conference of Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, IEEE, Nov. 2017.

[52] A. Shostack, "Threat modeling: Designing for security", Wiley (2014)

[53] G. Falco, A. Viswanathan, C. Caldera and H. Shrobe, "A Master Attack Methodology for an AI-Based Automated Attack Planner for Smart Cities", *IEEE Access*, Vol.6, pp.48360-48373 (2018)

[54] 萱島：「IoT システム向けセキュリティ要件定義手法の提案」, *電子情報通信学会論文誌 A*, 第 99 巻,

第 2 号，74-82 頁，2016 年 2 月

[55] JASE, "JASO, TP15002: Guideline for Automotive Information Security Analysis," Jan. 2016.

[56] J. Son, J. Kim, H. Na and D. Baik, "CBDAC: Context-Based Dynamic Access Control Model Using Intuitive 5W1H for Ubiquitous Sensor Network," *International Journal of Distributed Sensor Networks*, Vol.11.9, 836546, Sep. 2015.

[57] L. Gallon and J. Bascou, "Using CVSS in attack graph," 2011 Sixth International Conference on Availability, *Reliability and Security*, pp. 59–66, Aug. 2011.

[58] Y. Kawanishi, H. Nishihara, D. Souma, H. Yoshida and Y. Hata, " A Comparative Study of JASO TP15002-Based Security Risk Assessment Methods for Connected Vehicle System Design," *Hindawi Security and Communication Networks*, Vol. 2019, Article ID 4614721, Feb. 2019.

[59] Y. Kawanishi, H. Nishihara, D. Souma, H. Yoshida and Y. Hata, "A Study on Quantitative Risk Assessment Methods in Security Design for Industrial Control Systems," *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing*, pp. 62-69, Aug. 2018.

[60] W. Wideł , Peetam Mukherjee and Mathias Ekstedt, "Security Countermeasures Selection Using the Meta Attack Language and Probabilistic Attack Graphs," *IEEE Access*, Volume 10, Aug. 2022.

[61] O. Stan, R. Bitton, M. Ezrets, M. Dadon, M. Inokuchi, Y. Ohta, T. Yagyu, Y. Elovici, and A. Shabtai, "Heuristic Approach for Countermeasure Selection Using Attack Graphs," *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*, Jun. 2021.

[62] L. Wang, A. Singhal and S. Jajodia, "Toward measuring network security using attack graphs," in the proceedings of *the 2007 ACM workshop on Quality of protectionOctober 2007 (QoP'07)*, Pages 49–54, Oct. 2007.

[63] Y. Fei, J. Ning and W. Jiang, "A quantifiable Attack-Defense Trees model for APT attack," Proceedings 2018 *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Pages 2303-2306, Dec. 2018.

[64] NIST CNSSI 4009-2015, "Committee on National Security Systems (CNSS) Glossary," https://rmf.org/wp-content/uploads/2017/10/CNSSI-4009.pdf, Accessed on 24th Apr. 2024.

[65] 大久保：「セキュリティ要求工学」, *安全工学*, Vol.54, No.6, pp.460-463 (2015)

[66] C. Haley, R. Laney, J. Moffett and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Transactions on Software Engineering*, Vol. 34, Issue 1, pp. 133–153, Jan. 2018

[67] ISA, "ISA/IEC 62443 Series of Standards," https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards, Accessed on 24th Apr. 2024.

[68] G. Stoneburner, C. Hayden and A. Feringa, "Engineering Principles for Information Technology Security (A Baseline for Achieving Security)," NIST Special Publication 800-27, Jul. 2001.

[69] IEC 62443-3-2:2020, "Security for industrial automation and control systems - Part 3-2: Security risk assessment for system design," IEC, Jun. 2020.

[70] J. Meloy and J. Hoffmann, "International Handbook of Threat Assessment Second Edition," *Oxford University Press*, Apr. 2021.

[71] ISO 31000:2018 "Risk management - Guidelines," ISO, Feb. 2018.

[72] Microsoft, "The STRIDE Threat Model," https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20), Accessed on 24th Apr. 2024.

[73] S. Adam, "Experiences Threat Modeling at Microsoft," https://adam.shostack.org/modsec08/ Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf, Accessed on 24th Apr. 2024.

[74] T. UcedaVélez, "T. Real World Threat Modeling Using the PASTA Methodology. Technical report," *OWASP AppSec EU 2012*, Jul. 2012.

[75] 足立：「モデリング法」, *計測と制御*, 第 42 巻, 第 4 号, 262-267 頁, 2003 年 4 月

[76] P. Chen, “The entity-relationship model - toward a unified view of data,” *ACM Transactions on Database Systems*, Vol. 1, Issue 1, pp. 9–36, Mar. 1976

[77] J. Rumbaugh, I. Jacobson and G. Booch, " The Unified Modeling Language Reference Manual (2nd Edition)," *The Addison-Wesley Object Technology Series*, Jan. 2004

[78] S. Friedenthal, A. Moore and R. Steiner, "A Practical Guide to SysML: The Systems Modeling Language 2nd Edition," The MK/OMG Press, Oct. 2011

[79] T. Gruber, "What is an Ontology," 1993. https://queksiewkhoon.tripod.com/ontology_01.pdf, Accessed on 24th Apr. 2024.

[80] T. Gruber and R. Thomas, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, Vol. 5, Num. 2, pp. 199–220, Jun. 1993.

[81] P. Křemen and Z. Kouba, "Ontology-driven information system design," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42.3: 334-344, Sep. 2011.

[82] M. Uschold, "Ontology-driven information systems: Past, present and future. In: Formal Ontology in Information Systems," IOS Press, pp.3-18, Jul. 2008.

[83] A. Wiebe and C. Chan, "Ontology driven software engineering," *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp.1-4, Oct. 2012

[84] JIS Z 8115:2000, 「デイペンダビリティ（信頼性）用語」, 日本工業規格(JIS), 2000 年 10 月

[85] L. Singels, C. Biebuyck and L. Malukele, "A Formal Concept Analysis Driven Ontology for ICS Cyberthreats," in *Southern African Conference for Artificial Intelligence Research (SACAIR) 2020 Proceedings: Knowledge Representation and Reasoning*, pp.247-263, Dec. 2020.

[86] M. Jarwar, J. Watson, U. Ani and S. Chalmers, "Industrial Internet of Things Security Modelling using Ontological Methods," in the proceeding of *the 12th International Conference on the Internet of Things (IoT'22)*, pp.163-170, Nov. 2022.

[87] Lockheed Martin: Cyber Kill Chain®, https://www.lockheedmartin. com/en-us/capabilities/cyber/cyber-kill-chain.html, Accessed on 24th Apr. 2024.

[88] D. Miessl: “Securing the Internet of Things: Mapping Attack Surface Areas Using the OWASP IoT Top 10”, *RSAConference2015*, ASD-T10, Apr. 2015

[89] MITRE Corp., CAPEC™, https://capec.mitre.org/, Accessed on 24th Apr. 2024.

[90] N. Matsumoto, J. Fujita, H. Endoh, T. Yamada, K. Sawada and O. Kaneko, "Asset Management Method of Industrial IoT Systems for Cyber-Security Countermeasures," *MDPI Information 2021*, Vol. 12, pp. 460, Nov.

2021. Available: https://doi.org/10.3390/ info12110460

[91] S. Caltagirone, A. Pendergast, and C. Betz, "The Diamond Model of Intrusion Analysis", *DTIC Document*, Technical Report, Jul. 2013.

[92] A. Schaft, "Achievable behaviors of general systems," Systems & Control Letters, vol. 49, pp. 141-149, Jun. 2003.

[93] MITRE Corp., ATT&CK®, https://attack.mitre.org/

[94] MITRE Corp., ATT&CK® for Enterprise, https://attack.mitre.org/matrices/enterprise/, Accessed on 24th Apr. 2024.

[95] MITRE Corp.: ATT&CK® for Industrial Control Systems, https://attack.mitre.org/matrices/ics/

[96] MITRE Corp., "Group: Dragonfly 2.0, Berserk Bear, DYMALLOY, "ATT&CK® for Industrial Control Systems, Available: https://collaborate. mitre.org/attackics/index.php/Group/G0006, Accessed on Nov. 2021.

[97] MITRE Corp., "Lazarus Group," ATT&CK®, https://attack.mitre.org/groups/G0032/, Accessed on 24th Apr. 2024.

[98] CISA: "Alert (AA21-131A) DarkSide Ransomware: Best Practices for Preventing Business Disruption from Ransomware Attacks," Available: https://www.cisa.gov/uscert/ncas/ alerts/aa21-131a, May 2021.

[99] S. Barnum, "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information Rxpression (STIX) Version 1.1," MITRE Corporation, Feb. 2012.

[100] D. Binaco, " The Pyramid of Pain," Enterprise Detection & Response Blog, Available: https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html, Mar. 2013, Update Jan. 2014.

[101] T. Cody, "A Layered Reference Model for Penetration Testing with Reinforcement Learning and Attack Graphs," *2022 IEEE 29th Annual Software Technology Conference (STC)*, pp. 41-50, Oct. 2022.

[102] C. Phillips and L. P. Swiler, "A graph-based system for network vulnerability analysis," *The 1998 Workshop on New Security Paradigms*, pp. 71–79, Jan. 1998.

[103] O. Sheyner, J. Haines, S. Jha, R. Lippmann and J. M. Wing, "Automated generation and analysis of attack graphs," *2002 IEEE Symposium on Security and Privacy*, pp. 273–284, May 2002.

[104] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, IEEE, pp. 121–130, Dec. 2006.

[105] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, Ali E. Abdallah: "Cyber threat intelligence sharing: Survey and research directions", *Computers & Security*, Vol.87 (2019)

[106] S. Shiva, S. Roy and D. Dasgupta: "Game theory for cyber security," *the Sixth Annual Workshop on Cyber Security and Information Intelligence Researc*h, Apr. 2010.

[107] 小倉，藤田，松本：「産業制御システム向けセキュリティ対策効率化のためのサイバー攻撃シナリオ生成手法」，*電気学会論文誌C*，144 巻，1 号，35-42 頁，2024 年 01 月

[108] IEC: " IEC 62443-2-1:2010 Industrial communication networks - Network and system security - Part 2-1: Establishing an industrial automation and control system security program,", Nov. 2010.

[109] P. Mell, K. Scarfone and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," Jan. 2007. Available: https://www.first.org/cvss/v2/cvss-v2-guide.pdf

[110] MITRE Corp. "Common Weakness Scoring System (CWSS™) Version 1.0.1," Sep.2014. Available:

https://cwe.mitre.org/cwss/ cwss_v1.0.1.html, Accessed on 24th Apr. 2024.

[111] NIST, "NATIONAL VULNERABILITY DATABASE," https://nvd.nist.gov/, Accessed on 24th Apr. 2024.

[112] M. Bode, S. Oluwadare, B. Alese and A. F. Thompson, "Risk analysis in cyber situation awareness using Bayesian approach," *2015 International Conference on Cyber Situational Awareness*, Data Analytics and Assessment (CyberSA), pp.1-12, Jun. 2015.

[113] M. Mohsin, M. U. Sardar, O. Hasan and Z. Anwar, "IoTRiskAnalyzer: A Probabilistic Model Checking Based Framework for Formal Risk Analytics of the Internet of Things," *IEEE Access*, Vol.5, pp.5494-5505, Apr. 2017.

[114] X. Ou, W. Boyer and S. Zhang, "MulVAL: A logic-based network security analyzer," *14th USENIX Security Symposium*, Aug. 2005.

[115] International Charter, "The Risk Equation," https://www.icharter.org/articles/risk_equation.html, Accessed on 24th Apr. 2024.

[116] E. Kost, "5 Step Guide: How to Perform a Cyber Risk Analysis," https://www.upguard.com/blog/ how-to-perform-a -cyber-risk-analysis, Accessed on 24th Apr. 2024.

[117] Y. Hashimoto, T. Toyoshima, S. Yogo, M. Koike, T. Hamaguchi, S. Jing and I. Koshijima, "Safety securing approach against cyber-attacks for process control system," *Computers & Chemical Engineering*, Vol. 57, pp. 181-186, Oct. 2013.

[118] S. Ali and R. W. Anwar, "Trused: a trust-based security evaluation scheme for a distributed control system," *Computers, Materials & Continua 2023*, Vol. 74, No.2, pp. 4381–4398, Oct. 2022.

[119] M. Battaglioni, G. Rafaiani, F. Chiaraluce and M. Baldi, "MAGIC: A Method for Assessing Cyber Incidents Occurrence," *IEEE Access*, Vol. 10, pp. 73458-73473, Jul. 2022.

[120] FIRST, "Common Vulnerability Scoring System," Available: https://www.first.org/cvss/, Accessed on Nov. 2021.

[121] D. Woods and L. Walter, "Reviewing Estimates of Cybercrime Victimisation and Cyber Risk Likelihood," *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW),* pp. 150-162, Jun. 2022.

[122] F. Cremer, B. Sheehan, M. Fortmann, A. Kia, M. Mullins, F. Murphy and S. Materne, "Cyber risk and cybersecurity: a systematic review of data availability," *Geneva Papers Risk and Insurance Issues and Practice*, Vol. 47, pp. 698-736, Feb. 2022.

[123] I. Agrafiotis, J. R. C. Nurse, M. Goldsmith, S. Creese and D. Upton, "A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate," *Journal of Cybersecurity*, Vol. 4, Issue 1, Oct. 2018.

[124] E. Haapamäki and J. Sihvonen, "Cybersecurity in accounting research," *The Managerial Auditing Journal*, Vol. 34, Issue 7, Jul. 2019.

[125] D. Tsuji, J. Fujita, N. Matsumoto, Y. Tamura, J. Doenhoff & T. Shigemoto, "3-layer modelling method to improve the cyber resilience in Industrial Control Systems," *SICE Journal of Control, Measurement, and System Integration*, 16:1, 63-74, DOI: 10.1080/18824889.2023.2177074, Feb. 2023.

[126] IPA：「制御システムのセキュリティリスク分析ガイド 第 2 版（2023 年 3 月版）」, https://www.ipa.go.jp/security/controlsystem/ssf7ph00000098vy-att/000109380.pdf (2023)

[127] G. McGraw, "Software security," *IEEE Security & Privacy*, Vol. 2, Issue 2, Pages 80-83, Aug. 2004

[128] ISO/IEC 27005:2018, "Information technology — Security techniques - Information security risk management," Edition 3, Jul. 2018

[129] IEC 62443-2-1:2010, "Industrial communication networks - Network and system security - Part 2-1: Establishing an industrial automation and control system security program," Edition 1, Nov. 2010.

[130] P. Katsumata, J. Hemenway and W. Gavins, "Cybersecurity risk management", *2010 - MILCOM 2010 Military Communications Conference*, Pages 890-895, Oct 2010.

[131] X. Wenjun and R. Lagerström, "Threat modeling - A systematic literature review," Computers & Security, Volume 84, Pages 53-69, Jul. 2019.

[132] L. Haley, J. Moffett and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Transactions on Software Engineering*, Vol. 34, Issue 1, pp. 133–153, Jan. 2018

[133] ICS-CERT, "Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies," https://www.cisa.gov/uscert/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf, Sep. 2016.

[134] Microsoft, "Create a diagram with crow's foot database notation," https://support.microsoft.com/en-au/office/create-a-diagram-with-crow-s-foot-database-notation-1ec22af9-3bd3-4354-b2b5-ed5752af6769, Accessed on 24th Apr. 2024.

[135] N. Poolsappasit, R. Dewri and I. Ray, "Dynamic Security Risk Management Using Bayesian Attack Graphs," in *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 1, pp. 61-74, Jun. 2011

[136] 橋本，越島：「プロセス制御系のサイバーセキュリティ対策の立案と評価」，ヒューマンファクターズ，Vol.19，No.1, pp.18-25 (2014)

[137] W. Vesely, F. Goldberg, N. H. Roberts and D. F. Haasl, “Fault tree handbook,” Nuclear Regulatory Commission, Jan. 1981

[138] H. Kanamaru, J. Fujita and T. Arai, "A Study on the Classification of OT Security Risk Mitigation Measures," in the proceedings of *SICE Annual Conference 2023,* pp.274-279, Sep. 2023

[139] G. Chu and A. Lisitsa, "Ontology-based Automation of Penetration Testing," in the proceedings of *The 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pp.713–720, Feb. 2020.

[140] P. Johnson, R. Lagerström and M. Ekstedt, "A Meta Language for Threat Modeling and Attack Simulations," in the proceedings of *the 13th International Conference on Availability, Reliability and Security (ARIS)*, No.38, pp.1-8, Aug. 2018.

[141] R. Khan, K. McLaughlin, D. Laverty and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in the proceedings of *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp.1-6, Sep. 2017.

[142] P. Johnson, A. Vernotte, M. Ekstedt and R. Lagerström, "pwnPr3d: An Attack-Graph-Driven Probabilistic Threat-Modeling Approach," *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pp.278-283, Aug. 2016.

[143] A. Ekelhart, E. Kiesling, B. Grill, C. Strauss and C. Stummer, "Integrating attacker behavior in IT security analysis: a discrete-event simulation approach," *Information Technology and Management*, Vol.16, pp.221-233,

Jun. 2015.

[144] A. Rahman, R. Mahdavi-Hezaveh and L. Williams, "A systematic mapping study of infrastructure as code research," *Information and Software Technology*, Vol.108 pp. 65-77, Apr. 2019.

[145] Plattform Industrie 4.0, "Details of the Asset Administration Shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02)," Published by The Federal Ministry for Economic Affairs and Climate Action in Germany (BMWK), May 2022.

[146] E. Tantik and R. Anderl, "Integrated Data Model and Structure for the Asset Administration Shell in Industrie 4.0," *the 27th CIRP Design 2017*, Vol. 60, pp. 86-91, May 2017.

謝辞

本研究の遂行にあたり，多大なご指導とご助言を賜りました金子 修 教授，澤田 賢治 准教授に深く感謝を申し上げます．また，研究の成果を本論に纏めるにあたり，様々な有益なご意見をくださった小木曽 公尚 教授，清 雄一 教授，および新 誠一 名誉教授にも，心より感謝を申し上げます．

本研究の遂行に必要なデータや情報収集，手法の設計や実装，結果の議論をさせて頂いた出向元の直属上司である（株）日立製作所 研究開発グループ 松本 典剛 氏，同僚の小倉 貴志 氏，そして辻 大輔氏にも多大な感謝を申し上げます．更に本研究成果に関し，産業応用面での議論につきまして，ご議論させて頂きました，（株）日立製作所 マネージド＆プラットフォームサービス事業部の大河内 一弥 氏，（株）日立ソリューションズ 坂本 篤郎 氏，そして矢沢 澄仁 氏にも多大な感謝を申し上げます．

制御システムのセキュリティという大きなトピックについて，出向元である（株）日立製作所 研究開発グループ コネクティブオートメーションイノベーションセンタ 自立制御研究部 CAA1 ユニットの同僚の皆様に加え，企業の枠を超えて議論させて頂いた計測自動制御学会 産業応用部門 産業ネットワーク・システム部会の（株）三菱電機 神余 浩夫 氏，（株）横河電機 新井 貴之 氏他，同部門の皆様にも感謝を申し上げます．

現出向先の所属でもあり，本研究活動にご理解とご配慮いただきました現上司でもある Hitachi America, Ltd., R&D Division, Sudhanshu Gaur 氏他，IoT Edge Lab の同僚の皆様にも深く感謝を申し上げます．また，博士課程での学術研究を応援して頂いた前部長の髙橋 絢也 氏，現部長の伊藤 誠也 氏にも，この場を借りて感謝を申し上げます．

また，博士課程への進学のきっかけを与えて頂いた（株）日立製作所 制御プラットフォーム統括本部 中野 利彦 氏，そして，進学を後押し頂いた田野 俊一 学長にも改めて感謝いたします．加えて，本研究テーマを始め，制御システムのセキュリティというテーマを研究する機会を与えていただいた故 山田 勉 氏に多大なる感謝をお伝え致します．

最後に，本論の完成に際し，日頃より理解頂き，精神的な支えとなってくれた家族，友人の皆様に深い感謝の意を表します．

敬具

2024 年 9 月

関連論文の印刷公表の方法及び時期

(1) 藤田，小倉，大河内，松本，澤田，金子：「Diamond model と攻撃状態に基づくサイバー攻撃シナリオ構造化表現モデル」，*電気学会論文誌 C*，142 巻，3 号，328-338 頁，2022 年 03 月 （第 3 章に関連）

(2) J. Fujita, T. Ogura, K. Okochi, N. Matsumoto, K. Sawada and O. Kaneko, "A Structured Cyber Attack Representation Model based on the Diamond Model and Adversary States," in *IEEE Access*, doi: 10.1109/ACCESS.2023.3343639, Dec. 2023 (Early Access)（第 3 章，第 4 章に関連）

(3) 藤田，辻，矢沢，坂本，澤田，金子：「攻撃キルチェーンと多層防御モデルに基づく産業制御システム向けモデルベースセキュリティ対策設計方式」，*システム制御情報学会論文誌*，37 巻，1 号，1-11 頁，2024 年 01 月 （第 5 章に関連）

本博士論文の参考論文

(4) D. Tsuji, J. Fujita, N. Matsumoto, Y. Tamura, J. Doenhoff & T. Shigemoto, "3-layer modelling method to improve the cyber resilience in Industrial Control Systems," *SICE Journal of Control, Measurement, and System Integration*, Vol. 16, Issue 1, pp.63-74, DOI: 10.1080/18824889.2023.2177074, Feb. 2023

(5) H. Kanamaru, J. Fujita, T. Arai, "A Study on the Classification of OT Security Risk Mitigation Measures," in *Proceedings of SICE Annual Conference 2023*, pp.274-279, Sep. 2023

(6) 小倉，藤田，松本：「産業制御システム向けセキュリティ対策効率化のためのサイバー攻撃シナリオ生成手法」，*電気学会論文誌 C*，144 巻，1 号，35-42 頁，2024 年 01 月