様式７号（第 12 条，第 31 条関係）
（様式７号）(Format No.7)　日本語版

学　位　論　文　要　旨
(Summary of the Doctoral Dissertation)

学位論文題目 (Dissertation Title)	Intrusion Prevention System with Automated Rule Generation Using Automata and Large Language Models from Proof-of-Concept Codes （概念実証コードを用いてオートマトンと大規模言語モデルによってルールを生成する侵入防止システム）
氏　名(Name)	山本 裕大

サイバーセキュリティにおける PoC（概念実証）コードは、脆弱性やそれらの悪用への対策を検討するための情報を提供することで、セキュリティベンダやプログラムの作成者に、脆弱性を修正するためのパッチ作成を促すという点で非常に有用である。しかし、攻撃者が PoC コードを実際の攻撃に悪用したり、脆弱性の詳細な情報が拡散したりして、攻撃を実行する難易度が低下するなどの危険性もある。

2021 年 12 月に公開された、ロギングライブラリである Apache Log4j の脆弱性である Log4Shell は、Log4j が多くのアプリケーションやサービスに影響を与えたことで有名である。しかしそれだけでなく、Log4Shell は PoC コードが攻撃の拡大に貢献し、IDS などによる防御側の対策の不十分さを明確に示した脆弱の 1 つでもある。

これまで、IDS や IPS のルール作成における各タスクは、人間による手作業を中心として行われており、1 つのルールを作成するために多くの時間を必要としている。ルールの作成に多くの時間を必要としていることは、Log4Shell のように脆弱性が公開されてから、脆弱性を悪用する攻撃が大規模に行われるまでの期間が短い状況において、脆弱性を悪用する攻撃が大規模に開始される前に、それらへの対策を講じることが難しいことを意味する。ルール生成を自動化することによって、ルールの作成にかかる時間を削減することができるが、これまでに行われた研究は汎用性や、複雑性を持つ攻撃への対処に課題がある。

そこで学位論文では、Log4Shell の攻撃パターンで使用される難読化法を定義した「事前知識」を使用することによって、難読化やランダム性といった複雑性を持つ攻撃パターンを対象としたルールを生成する手法と、脆弱性の攻撃に関する有用な情報を得られる PoC コードの内容を自動的に分析することによって、ルールの作成に要する時間を最小化する手法を提案した。

本論文は、5 つの章から構成される。

第 1 章では、事前知識を用いたルール生成手法によって、難読化やランダム性を持つ攻撃パターンに効果的に対処し、LLM による自動的なルール生成手法によって、脆弱性を悪用する攻撃が大規模に開始される前にそれらに対処することを可能にすることで、人間によるルール作成を助けるという、本論文の目的と解決する課題を示した。

第 2 章では、本論文で提案する 2 つのルール生成手法の必要性を明確にするために、Log4Shell や IPS・IDS で使用されるルールについて説明し、Log4Shell に対抗する手法や、IPS や IDS に適用可能なルールを自動的に生成する手法とそれらの課題を示し、これまでに提案された手法は、Log4Shell の難読化やランダム性を持つ攻撃パターンに対して、それらを検出するルールを生成することが難しいことを明らかにした。

第 3 章では、Log4Shell の攻撃パターンで使用される難読化法を定義した事前知識と攻撃パターンとの、オートマトンを用いたマッチングによってルールを生成する手法を提案した。この手法は、

（和文 2,000 字程度 / 英文 800 語程度）
(about 800 words)

Log4Shell の悪用に用いられる攻撃パターンの、難読化とランダム性に効果的に対処することを可能にし、生成されたルールの可視化は、生成されたルールとルール生成に使用された事前知識の間の関係性を明確化し、生成されたルールの検出対象の理解を容易にする。提案手法の有効性を検証する実験では、既存のルールセットに含まれる、Log4Shell を対象としたルールと、提案手法によって生成されたルールを比較した。比較によって、提案手法によって生成されたルールは、事前知識として与えられた難読化法のみが用いられた攻撃パターンを使用した場合において、比較対象のルールよりも 60%高い 100%の検出性能を示した。また、提案手法のルールは同じ難読化法に対して、比較対象のルールよりも、62%短い長さのルールを生成できることを示し、提案手法は Log4Shell の攻撃パターンの難読化やランダム性に、効果的に対処できることを明らかにした。

第 4 章では、PoC コードの内容を LLM によって分析することで、IDS 向けのルールを自動的に生成する手法を提案した。この手法は、人間によるルール作成において多くの時間を必要とする、脆弱性の分析を含めた、ルール作成の一連のタスクを自動化する。提案手法の有効性を検証する実験では、プログラミング言語やフォーマットなどが異なる複数の PoC コードを用いて、生成されたルールの検出性能や、ルールの生成に要した時間を比較した。Log4Shell を悪用する攻撃で用いられる攻撃パターンを検出可能なルールの生成に成功し、いずれの PoC コードを使用した場合も、ルールを 1 分以内に作成するとともに、人間による最小限の修正のみで実用的なルールを生成できることが示された。これらの結果は、提案手法が PoC コードで用いられるプログラミング言語やフォーマットに依存しない汎用性を持ち、新たな攻撃パターンの登場にすぐに対処できることを明らかにした。

第 5 章では、結論として本論文で提案した手法が Log4Shell の攻撃パターンに対して有効なルールを生成できることを改めて説明した上で、マルウェアの検出への応用可能性や、学位論文で提案した 2 つの手法を組み合わせることについても論じている。

様式７号（第 12 条，第 31 条関係）
（様式７号）(Format No.7)　英語版

学　位　論　文　要　旨
(Summary of the Doctoral Dissertation)

学位論文題目 (Dissertation Title)	Intrusion Prevention System with Automated Rule Generation Using Automata and Large Language Models from Proof-of-Concept Codes （概念実証コードを用いてオートマトンと大規模言語モデルによってルールを生成する侵入防止システム）
氏　名(Name)	YAMAMOTO Yudai

Proof-of-concept (PoC) code in cybersecurity is very useful in that it provides information to consider vulnerabilities and countermeasures against their exploitation, encouraging security vendors and program creators to create patches to fix vulnerabilities. However, there is also the risk that attackers may misuse PoC code in actual attacks or detailed information about the vulnerability may spread, decreasing the difficulty of carrying out attacks.

Log4Shell, a vulnerability in the Apache Log4j logging library that was disclosed in December 2021, is well-known for affecting many applications and services caused by Log4j. However, Log4Shell is also one of the vulnerabilities in which PoC code contributed to the spread of attacks and clearly demonstrated the inadequacy of defensive measures such as IDS.

Until now, each task in creating IDS and IPS rules has been mainly performed manually by humans, and it takes a lot of time to create one rule. The fact that it takes a lot of time to create rules means that it is difficult to take measures before large-scale attacks exploiting vulnerabilities begin in a situation where the time between the disclosure of vulnerabilities and large-scale attacks exploiting the vulnerabilities is short, as in the case of Log4Shell. Although the time required to create rules can be reduced by automating rule generation, research conducted to date has issues with versatility and dealing with complex attacks.

In this thesis, we proposed a method to generate rules targeting complex attack patterns such as obfuscation and randomness by using "prior knowledge" that defines the obfuscation method used in Log4Shell attack patterns, and a method to minimize the time required to create rules by automatically analyzing the contents of PoC code that can obtain useful information about vulnerability attacks.

This thesis consists of five chapters.

Chapter 1 describes the objectives of this paper and the problems to be solved: to effectively deal with obfuscated and random attack patterns using a rule generation method that uses prior knowledge, and to assist human rule creation by enabling attacks that exploit vulnerabilities to be dealt with before they are launched on a large scale using an automatic rule generation method using LLM.

Chapter 2 explains the rules used in Log4Shell and IPS/IDS to clarify the necessity of the two rule generation methods proposed in this paper, and presents methods to counter Log4Shell and methods to automatically generate rules applicable to IPS and IDS, as well as the challenges they pose. It is clear that the methods proposed so far have difficulty generating rules to detect

（和文 2,000 字程度 / 英文 800 語程度）
(about 800 words)

obfuscated and random attack patterns in Log4Shell.

Chapter 3 proposes a method to generate rules by using automata to match attack patterns with prior knowledge that defines the obfuscation methods used in Log4Shell attack patterns. This method makes it possible to effectively deal with the obfuscation and randomness of attack patterns used to exploit Log4Shell, and visualization of the generated rules clarifies the relationship between the generated rules and the prior knowledge used to generate the rules, making it easier to understand what the generated rules are intended to detect. In an experiment to verify the effectiveness of the proposed method, rules targeting Log4Shell included in an existing rule set were compared with the rules generated by the proposed method. The comparison showed that the rules generated by the proposed method showed 100% detection performance, 60% higher than the comparison rules, when an attack pattern was used that used only the obfuscation method given as prior knowledge. In addition, it was shown that the rules of the proposed method can generate rules that are 62% shorter in length than the comparison rules for the same obfuscation method, making it clear that the proposed method can effectively deal with the obfuscation and randomness of Log4Shell attack patterns.

In Chapter 4, we proposed a method to automatically generate rules for IDS by analyzing the contents of PoC code using LLM. This method automates a series of tasks in rule creation, including vulnerability analysis, which requires a lot of time when creating rules by humans. In an experiment to verify the effectiveness of the proposed method, we used multiple PoC codes with different programming languages and formats to compare the detection performance of the generated rules and the time required to generate the rules. We succeeded in generating rules that can detect attack patterns used in attacks that exploit Log4Shell, and it was shown that for each PoC code, rules could be created within one minute and practical rules could be generated with only minimal human modification. These results clarified that the proposed method is versatile and does not depend on the programming language or format used in the PoC code, and can quickly respond to the emergence of new attack patterns.

In Chapter 5, we conclude by explaining that the method proposed in this paper can generate effective rules for Log4Shell attack patterns, and also discuss the possibility of applying it to malware detection and the combination of the two methods proposed in the doctoral thesis.

（様式９号）

学位論文審査の結果及び最終試験の結果報告書

山口大学大学院創成科学研究科

氏　　名	山本　裕大
審査委員	主　　査：山口　真悟
	副　　査：多田村　克己
	副　　査：福士　　将
	副　　査：中正　和久
	副　　査：井田　悠太
論文題目	Intrusion Prevention System with Automated Rule Generation Using Automata and Large Language Models from Proof-of-Concept Codes（概念実証コードを用いてオートマトンと大規模言語モデルによってルールを生成する侵入防止システム）

【論文審査の結果及び最終試験の結果】

本学位論文では、サイバー攻撃からシステムを保護するための侵入防止システム（IPS）のルールを自動生成する手法を提案し、特にルール作成の自動化と品質向上に焦点を当てている。従来、IPS のルールは専門家が手作業で作成していたため、時間と労力を要し、新たな攻撃手法の出現に迅速に対応することが困難であった。そこで本論文では、IPS のルールを自動生成する二つの新たな手法を提案し、その有効性を検証した。

一つ目の手法は、事前知識とオートマトンを活用したルール生成である。この手法では、攻撃パターンで使用される難読化法を定義した「事前知識」と、オートマトンと呼ばれる数学的かつグラフィカルなモデルを組み合わせることで、新たな攻撃パターンを予測し、IPS のルールを自動生成する。このアプローチにより、人間が理解しやすいルールを生成し、より効率的な攻撃検出を実現することが可能となる。実験結果では、既存のルールと比較して検出性能が 60%向上した。さらに、同じ難読化法に対して、既存のルールよりも 62%短い長さのルールを生成できることが確認された。

二つ目の手法は、LLM（大規模言語モデル）を用いた PoC（概念実証）コード分析によるルール生成である。PoC コードは、脆弱性の悪用方法を示すサンプルコードであり、攻撃者が悪用するリスクがある一方で、防御側が迅速に対応するための有用な情報も提供する。従来、PoC コードの分析とルール作成は人間による手作業で行われており、多くの時間を要していた。そこで、LLM を用いて PoC コードの内容を自動的に分析し、IPS 向けのルールを生成する手法を提案した。この手法は、脆弱性の分析を含めたルール作成の一連のタスクを自動化する。実験結果では、実際の PoC コードを用いて、ルールを 1 分以内に生成することに成功した。また、表現形式が異なる PoC コードに対しても、ルールを生成できることが確認

された。

本論文で提案した二つの手法は、IPSのルール作成における人的な負担を軽減し、より迅速かつ正確なルール生成を実現するものである。これらの手法は、IPSのルール作成における革新的なアプローチであり、その成果はその有効性を裏付けるものである。

公聴会における主な質問内容は、提案手法の利用者と使い分けに関するもの、提案手法の検出率に関するもの、今後の展望に関するものなどについてであった。いずれの質問に対しても発表者から的確な回答がなされた。

以上より本研究は独創性、信頼性、有効性、実用性ともに優れ、博士（工学）の論文に十分値するものと判断した。

論文内容及び審査会、公聴会での質問に対する応答などから、最終試験は合格とした。

なお、主要な関連論文の発表状況は下記のとおりである。（関連論文 計4編、参考論文 計0編）

1) Yudai Yamamoto, Shingo Yamaguchi, “Defense Mechanism to Generate IPS Rules from Honeypot Logs and Its Application to Log4Shell Attack and Its Variants,” Electronics, vol.12, no,14, 3177, 2023.
2) Yudai Yamamoto, Aoi Fukushima, Shingo Yamaguchi, “Implementation of White-Hat Worms Using Mirai Source Code and Its Optimization through Parameter Tuning,” Future Internet, vol.16, no.9, 336, 2024.
3) Yudai Yamamoto, Shingo Yamaguchi, “A Method to Prevent Known Attacks and Their Variants by Combining Honeypots and IPS,” Proceedings of 2022 IEEE 11th Global Conference on Consumer Electronics, pp.302-305, 2022.
4) Yudai Yamamoto, Shingo Yamaguchi, “On an LLM-Based Method to Generate from PoC Codes to IPS/IDS Rules,” Proceedings of International Conference on Electronics, Information, and Communication 2025, pp.273-277, 2025.