Computational Analysis of Security Detection Rules: Methodologies for Similarity Detection and Classification

Abstract

This paper presents a comprehensive methodology for analyzing and classifying security detection rules from multiple repositories, specifically focusing on rules from SigmaHQ, Splunk, and Elastic. We introduce a multi-dimensional approach that combines clustering, similarity analysis, and topic modeling to identify related rules across disparate sources. Our methodology leverages natural language processing (NLP) and machine learning techniques to extract meaningful patterns from rule contents, enabling security professionals to understand similarities, differences, and coverage gaps across detection rule ecosystems. Experimental results demonstrate the effectiveness of our approach in identifying functionally similar rules despite syntactic differences, revealing thematic groupings across repositories, and mapping rules to established security frameworks.

1. Introduction

Security detection rules play a critical role in identifying malicious activity across enterprise environments. Various open-source and commercial entities have developed extensive collections of detection rules, each with their own format, structure, and focus. The heterogeneity of these rule repositories presents challenges for security teams attempting to understand their overall detection coverage, identify redundancies, and discover gaps.

This paper addresses three fundamental research questions:

1. How can we identify functionally similar detection rules across repositories with different formats and syntactical structures?
2. What computational methods can effectively extract thematic patterns from large collections of security rules?
3. How can we organize detection rules into meaningful groups that facilitate better understanding of the security landscape?

We present a systematic framework that combines various computational methods to analyze and categorize security detection rules. Our approach extends beyond simple textual similarity to incorporate semantic understanding of rule descriptions, detection logic, and taxonomic classifications.

2. Related Work

2.1 Text Similarity and Clustering

Text similarity measures have been extensively studied in information retrieval and natural language processing. Vijaymeena and Kavitha [1] provide a comprehensive survey of document similarity metrics, including cosine similarity, Jaccard similarity, and embedding-based approaches. In the security domain, Guo et al. [2] applied similar techniques to cluster malware samples based on behavioral patterns.

2.2 Topic Modeling for Security

Topic modeling has been applied to various security-related texts, including vulnerability descriptions [3], security bulletins [4], and malware analysis reports [5]. These approaches typically employ Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF) to identify latent topics in document collections.

2.3 Security Rule Analysis

While substantial research exists on analyzing network traffic patterns and security logs, relatively little work has focused specifically on the analysis of security detection rules themselves. Harang and Kott [6] explored methods for evaluating the effectiveness of detection rules but did not address the problem of cross-repository rule similarity.

3. Dataset Description

Our study examines security detection rules from three major repositories:

1. **SigmaHQ**: An open-source detection rule format with a focus on SIEM-agnostic rules
2. **Splunk Security Content**: Detection rules specifically designed for Splunk environments
3. **Elastic Detection Rules**: Rules tailored for the Elastic Security platform

Each repository follows different formatting conventions:

- SigmaHQ uses YAML with a standardized Sigma format
- Splunk Security Content uses YAML with Splunk-specific search syntax
- Elastic Detection Rules uses TOML with Elasticsearch query DSL

The repositories collectively contain thousands of rules covering various security domains, including endpoint detection, network monitoring, cloud security, and threat hunting.

4. Methodology

Our methodology consists of four main components, each addressing a different aspect of rule analysis:

4.1 Rule Extraction and Preprocessing

We developed a unified parsing framework to extract structured information from the heterogeneous rule formats. For each rule, we extract:

- Title and description
- Detection logic (normalized to a standardized representation)
- Tags and metadata
- Source repository

Text preprocessing includes:

- Tokenization and lemmatization
- Removal of stop words
- Special character normalization
- Field-specific weighting (title, description, detection logic)

4.2 Vector Representation and Similarity Computation

We employ Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to transform rule texts into numerical representations. The TF-IDF value for a term t in document d from a corpus D is computed as:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Where:

$$\text{TF}(t, d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$$

$$\text{IDF}(t, D) = \log \frac{\text{Total number of documents in corpus D}}{\text{Number of documents containing term t}}$$

We calculate pairwise cosine similarity between rule vectors. For rule vectors A and B, the cosine similarity is:

$$\text{similarity}(A, B) = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

This produces a similarity matrix S where element S_{ij} represents the similarity between rules i and j.

4.3 Clustering and Group Detection

We implement two complementary approaches for grouping related rules:

4.3.1 K-means Clustering

The k-means algorithm partitions the rule vectors into k clusters by minimizing the within-cluster sum of squares:

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2$$

Where C_i is the i-th cluster and $\boldsymbol{\mu}_i$ is the centroid of cluster C_i.
The number of clusters k is determined experimentally, balancing cohesiveness and interpretability.

4.3.2 Threshold-based Similarity Grouping

We construct an undirected graph $G = (V, E)$ where vertices V represent rules, and an edge $e_{ij} \in E$ exists if $S_{ij} \geq \tau$, where τ is a similarity threshold.
Connected components in this graph form natural groups of similar rules, allowing for more fine-grained similarity detection than k-means clustering.

4.4 Topic Modeling and Thematic Analysis

We employ Non-negative Matrix Factorization (NMF) for topic modeling, which approximates the TF-IDF matrix V as the product of two non-negative matrices:

$$V \approx WH$$

Where W is the document-topic matrix and H is the topic-term matrix.

For a corpus with m documents and n terms, NMF finds matrices W of shape $m \times k$ and H of shape $k \times n$ that minimize the objective function:

$$||V - WH||_F^2$$

Where $||\cdot||_F$ is the Frobenius norm and k is the number of topics.

4.5 MITRE ATT&CK Framework Mapping

We leverage existing MITRE ATT&CK tags within the rules to map them to standard tactics and techniques. For each rule, we extract:

- Associated MITRE tactics (e.g., "initial_access", "execution")
- Associated MITRE techniques (e.g., "T1566", "T1059")

This mapping enables analysis of detection coverage across the MITRE ATT&CK matrix.

5. Dimensionality Reduction and Visualization

To enable intuitive visualization of rule similarities, we apply t-Distributed Stochastic Neighbor Embedding (t-SNE) to project the high-dimensional rule vectors into a two-dimensional space.

The t-SNE algorithm minimizes the Kullback-Leibler divergence between two probability distributions: one representing pairwise similarities in the high-dimensional space and another in the low-dimensional embedding space.

For points x_i and x_j in the high-dimensional space, the similarity is modeled as a conditional probability:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2 / 2\sigma_i^2)}$$

And for points y_i and y_j in the low-dimensional space:

$$q_{j|i} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq i} (1 + ||y_i - y_k||^2)^{-1}}$$

The algorithm minimizes:

$$\mathrm{KL}(P||Q) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

This projection preserves local similarities while revealing global structure in the rule collection.

6. Experimental Results

6.1 Dataset Characteristics

The analyzed dataset contains:

- 1,782 rules from SigmaHQ (Windows subset)
- 943 rules from Splunk Security Content (endpoint subset)
- 576 rules from Elastic Detection Rules (Windows subset)

6.2 Clustering Results

K-means clustering with $k=30$ produced coherent rule groupings. Key clusters include:

1. **Process Creation Monitoring**: Rules detecting suspicious process creation events
2. **Registry Manipulation**: Rules focusing on registry modifications
3. **PowerShell Abuse**: Rules detecting malicious PowerShell usage
4. **Credential Access**: Rules related to credential theft and manipulation
5. **Network Connection Monitoring**: Rules detecting suspicious network activities

The distribution of rules across repositories varied significantly by cluster, revealing differences in coverage focus.

6.3 Similarity Network Analysis

The threshold-based similarity grouping ($\tau = 0.7$) identified 219 groups of highly similar rules, with 658 rules (20% of the total) belonging to at least one similarity group.

Notable findings include:

- 127 groups containing rules from multiple repositories, indicating functional overlap
- 92 groups containing rules from a single repository, suggesting unique coverage areas

The largest similarity group contained 12 rules across all three repositories, all relating to detecting Mimikatz credential theft.

6.4 Topic Modeling Insights

NMF topic modeling with 15 topics revealed thematic patterns across repositories. The most prominent topics include:

1. **Living-off-the-Land Techniques**: Use of built-in Windows tools for malicious purposes
2. **Defense Evasion**: Techniques for bypassing security controls
3. **Ransomware Indicators**: Signs of ransomware activity
4. **Command and Control Communication**: Malicious network traffic patterns
5. **Privilege Escalation**: Techniques for gaining higher privileges

6.5 MITRE ATT&CK Coverage

Analysis of MITRE ATT&CK mappings revealed:

- "Execution" was the most frequently covered tactic (28% of all rules)
- "Defense Evasion" was the second most common (23% of all rules)
- T1059 (Command and Scripting Interpreter) was the most frequently covered technique
- T1055 (Process Injection) showed the most significant variation in coverage across repositories

7. Discussion

7.1 Cross-Repository Rule Similarity

Our analysis demonstrates significant overlap in detection capabilities across repositories, despite differences in syntax and format. Approximately 20% of rules have functionally similar counterparts in other repositories, suggesting opportunities for consolidation and standardization.

The similarity patterns also reveal specialization areas within each repository:

- SigmaHQ provides broader coverage across diverse detection areas
- Splunk Security Content shows strength in PowerShell and Windows event monitoring
- Elastic Detection Rules demonstrates robust coverage of process creation and injection techniques

7.2 Thematic Trends in Detection Logic

Topic modeling revealed consistency in thematic focus across repositories, with emphasis on script-based attacks, credential theft, and defense evasion techniques. These alignments reflect the current threat landscape, where attackers increasingly utilize built-in operating system features to evade detection.

The distribution of topics also highlights temporal shifts in detection priorities, with newer rules focusing more on cloud security, container orchestration, and identity-based attacks.

7.3 Coverage Gaps and Recommendations

By mapping rules to the MITRE ATT&CK framework, we identified several areas with limited detection coverage:

- Impact tactics (beyond ransomware)
- Exfiltration techniques
- Hardware-based attacks
- Social engineering detection

These gaps represent opportunities for security teams to enhance their detection capabilities through targeted rule development.

8. Limitations and Future Work

While our approach demonstrates effectiveness in analyzing security detection rules, several limitations should be addressed in future work:

1. **Semantic Understanding**: Our current methods rely primarily on lexical similarity rather than deep semantic understanding of detection logic.
2. **Rule Effectiveness**: Our analysis focuses on rule similarity but does not assess the effectiveness or accuracy of the rules themselves.
3. **Temporal Evolution**: The current study provides a static snapshot without accounting for the evolution of detection techniques over time.

Future work will explore:

- Incorporating detection logic parsing for more precise functional similarity assessment
- Developing methods to evaluate rule quality and effectiveness
- Implementing continuous analysis to track the evolution of detection rule repositories
- Extending the analysis to include additional rule repositories and formats

9. Conclusion

This paper presents a comprehensive methodology for analyzing security detection rules across repositories with different formats and structures. By combining clustering, similarity analysis, and topic modeling, we provide insights into the relationships and patterns within the broader detection rule ecosystem.

Our approach enables security professionals to:

1. Identify functionally equivalent rules across repositories
2. Understand thematic patterns in detection logic
3. Assess detection coverage against security frameworks
4. Discover potential gaps in detection capabilities

These insights can inform more efficient rule management, improved detection coverage, and better alignment of security monitoring with the evolving threat landscape.

References

[1] Vijaymeena, M. K., & Kavitha, K. (2016). A survey on similarity measures in text mining. Machine Learning and Applications: An International Journal, 3(2), 19-28.

[2] Guo, Y., Sima, C., Murphey, Y. L., & Qiu, S. (2018). A study of malware clustering based on behavioral signatures. In 2018 International Conference on Security and Privacy in Communication Networks.

[3] Neuhaus, S., & Zimmermann, T. (2010). Security trend analysis with CVE topic models. In 2010 IEEE 21st International Symposium on Software Reliability Engineering.
[4] Mulwad, V., Li, W., Joshi, A., Finin, T., & Viswanathan, K. (2011). Extracting information about security vulnerabilities from web text. In Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.
[5] Lim, S. K., Muis, A. O., Lu, W., & Ong, C. H. (2017). MalwareTextDB: A database for annotated malware articles. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.
[6] Harang, R. E., & Kott, A. (2017). Metrics of effectiveness for computer network defense security controls. In 2017 IEEE Conference on Communications and Network Security (CNS).
[7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[8] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788-791.
[9] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11), 2579-2605.
[10] Strom, B. E., Battaglia, J. A., Kemmerer, M. S., Kupersanin, W., Miller, D. P., Wunder, C., Whitley, S. D., Wolf, R. D. (2017). Finding cyber threats with ATT&CK-based analytics. The MITRE Corporation Technical Report.