




Report 123

Threat Detection from Sysmon Logs Using Machine Learning and Cyber Kill Chain Modeling.docx

-  My Files
-  My Files
-  University

Document Details

Submission ID

trn:oid::17268:93240064

Submission Date

Apr 28, 2025, 9:36 AM GMT+5:30

Download Date

Apr 28, 2025, 9:43 AM GMT+5:30

File Name

Threat Detection from Sysmon Logs Using Machine Learning and Cyber Kill Chain Modeling.docx

File Size

779.8 KB

9 Pages**5,050 Words****30,193 Characters**





5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

-  **22 Not Cited or Quoted 4%**
Matches with neither in-text citation nor quotation marks
-  **2 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 1%  Internet sources
- 2%  Publications
- 4%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 22 Not Cited or Quoted 4%**
Matches with neither in-text citation nor quotation marks
- 2 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 1% Internet sources
- 2% Publications
- 4% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	BB9.1 PROD on 2025-04-10	<1%
2	Submitted works	The University of the West of Scotland on 2023-07-02	<1%
3	Internet	edr.tymyrdin.dev	<1%
4	Submitted works	EC-Council University on 2025-04-17	<1%
5	Internet	www.bartleby.com	<1%
6	Internet	www.ncbi.nlm.nih.gov	<1%
7	Submitted works	Belgium Campus iTiversity NPC on 2024-06-20	<1%
8	Submitted works	National Institute of Business Management Sri Lanka on 2020-08-14	<1%
9	Submitted works	University of Greenwich on 2025-04-22	<1%
10	Submitted works	The University of the West of Scotland on 2024-03-27	<1%

11	Submitted works	American Public University System on 2023-11-13	<1%
12	Publication	Potts, Philip Lynn. "Action-A Model for Prioritizing Cross Sector Aviation Cyber Th...	<1%
13	Publication	"Advanced Information Networking and Applications", Springer Science and Busi...	<1%
14	Publication	Durgesh Kumar Mishra, Nilanjan Dey, Bharat Singh Deora, Amit Joshi. "ICT for Co...	<1%
15	Publication	Florian Skopik. "Collaborative Cyber Threat Intelligence - Detecting and Respondi...	<1%
16	Publication	Giovanni Paolo Carlo Tancredi, Luca Preite, Giuseppe Vignali. "Digital twin enhan...	<1%
17	Submitted works	University of Carthage on 2024-09-09	<1%
18	Submitted works	University of Queensland on 2024-10-15	<1%
19	Internet	link.springer.com	<1%
20	Internet	techscience.com	<1%
21	Submitted works	Boise State University on 2023-07-24	<1%
22	Submitted works	Napier University on 2015-08-14	<1%

Threat Detection from Sysmon Logs Using Machine Learning and Cyber Kill Chain Modeling

Abstract—In the current era of cybersecurity, detecting sophisticated threats at an early stage is increasingly critical. Traditional detection mechanisms, such as signature-based antivirus systems, struggle against evolving tactics, techniques, and procedures (TTPs) used by adversaries. Sysmon logs, offering rich and granular insights into system-level activities, represent a valuable data source for advanced threat detection. This research proposes an integrated framework that combines Sysmon log analysis with machine learning algorithms, while modeling attack behaviors according to the Cyber Kill Chain (CKC) methodology. The method establishes correlations between system events and different stages of CKC which helps in both situating activity behavior and predicting cyber intrusions. The research team conducted systematic work on Sysmon telemetry features development while performing model optimization experiments and validating the approach through real-world data analysis. The implementation of behavioral context using the Kill Chain model leads to enhanced machine learning model efficiency which results in detection rates surpassing 96%. The following paper explores a thorough discussion about the proposed approach while assessing its performance boundaries along with recommendations for upcoming research work. This research demonstrates the necessity of domain awareness integration into data-oriented cybersecurity systems which creates foundations for next-generation security threat detection technologies.

The document uses following terms— Sysmon Logs, Cybersecurity, Machine Learning, Cyber Kill Chain, Threat Detection, Behavioral Analysis, Advanced Persistent Threats (APT).

The system includes three important components: Advanced Persistent Threat (APT), Anomaly Detection, Attack Lifecycle Modeling.

I. INTRODUCTION

The occurrence of cyberattacks has skyrocketed throughout recent years putting organizations governments and individual worldwide at great danger. Modern cybercriminals build their methods with increasingly advanced techniques in order to successfully breach digital infrastructure networks. These criminal activities consist of Data Breaches as well as Denial-of-Service (DoS) attacks and the more sophisticated long-term Advanced Persistent Threats (APTs). Advanced Persistent Threats represent a particular attack strategy which enables hackers to quietly penetrate networks while staying unnoticed for long periods until they successfully execute their goals without incident detection. APT attacks differ from standard digital intrusions because they use an evolving sequence of stealthy tactics which outlast typical attacks that display obvious malware or exploits.

Traditional defenses like firewalls alongside antivirus software alongside signature-based intrusion detection systems become less effective at protecting networks against modern complex threats due to the modern systems' complexity alongside increased number of network devices.

Traditional protective measures depend on fixed markers and pre-written protocols that attackers achieve bypass with manipulative methods as well as variant malware programs and unreported system weaknesses. Most organizations struggle to defend against contemporary cyber threats which persist undetectably through their systems.

Need for Early Detection in Cybersecurity

Security measures for protecting networks depend fundamentally on detecting hostile actions as early as possible. The time security teams receive before an attack occurs directly corresponds with their ability to respond and minimize resulting damage. These considerations become vital for detecting APTs as these cyberattacks specifically target standard detection systems during their protracted phases of reconnaissance and infiltration and exfiltration before execution.

The analysis of system behavior represents a promising method for detecting attacks at an early stage. The methodology of behavioral analytics assumes no knowledge of attack signatures while it investigates system behavior together with user actions and network traffic monitoring to detect abnormal patterns linked to malicious events. System activity reveals detailed insights through Sysmon logs because they record complete information about system actions to demonstrate process and network connection and file system behavioral patterns. System logs enable deep data analysis for identifying unknown malicious activities along with identified suspicious operations.

1.1 Challenges with Raw Telemetry Data

The information Sysmon logs supply to analysts creates substantial difficulties for analysts to process. The streaming data contains multiple dimensions along with large amounts of noise which includes numerous irrelevant and redundant properties without significance toward threat detection. Lack of semantic understanding within the data makes it difficult to determine whether a network connection is dangerous since it doesn't recognize the relationship between initiating processes whether they are trusted or compromised ones.

Multiple ML techniques work together to examine enormous high-dimensional datasets to discover hidden patterns that exceed human analysts' perceptual capabilities. ML models can separate typical system activities from irregular ones after studying both normal user actions and security violations. Sophisticated Machine Learning models tend to make multiple incorrect alerts coupled with attacks which escape detection when models lack enough understanding of the whole system context. The detection performance degrades significantly when datasets contain few malicious logs compared to benign logs.

1.2 Proposed Solution: ML and Cyber Kill Chain (CKC) Modeling

The proposed solution combines Machine Learning techniques with Cyber Kill Chain (CKC) framework. Lockheed Martin created the Cyber Kill Chain as a systematic approach that divides attacks into sequential logical phases. These stages include:

1. Gathering information: The attacker gets hold of information about the target, its IP addresses, vulnerabilities, and the network configuration.
2. Attacker weaponization: The attacker produces weaponized payload or exploit to send to the target.
3. Payload: The payload is what the attacker is able to deliver.
4. Payload: What the attacker has that he uses to exploit the vulnerability in the system to run.
5. The attacker installs malicious software to ensure that the software remains running even after the attacker has left the system.
6. Attacker establishes communication channel to control the compromised system (C&C).
7. The final objectives of the attack: Actions on Objectives: The attacker executes the last objectives of his attack by exfiltrate data, destroy the system, or do the lateral movement.

By standing in as a useful lens to understand and model the stages of a cyberattack, security teams and security professionals can now anticipate and predict where the attackers will move in a network. If we can associate Sysmon logs to CKC stages, we can provide a more structured and contextualized way of detecting threats.

1.3 Machine Learning and Cyber Kill Chain Integration

In our approach we map Sysmon to the different stages of the Cyber Kill Chain to provide a framework to capture the development of an attack from beginning to end. In this approach, ML algorithms are used to examine the given data for any patterns in the data in order to find anomalous activities which correspond to stages of an attack. For instance, we can use Random Forests, XGBoost, or Logistic Regression to map out transition from any one stage of the kill chain to another. These models are trained on historical data with labels against CKC stages, the training then performs prediction about whether a new log is benign or potentially an attack. [1] , [7]

This hybrid approach takes advantage of several:

- Proactive Detection: By mapping behavior to CKC stages, the model can detect attacks earlier in the kill chain, reducing the potential damage.
- Context-Aware Detection: The integration of CKC stages provides context to the raw telemetry data, making it harder for attackers to obfuscate their actions.
- Enhanced Accuracy: The combined approach improves accuracy and reduces false positives by incorporating temporal context and multi-stage attack detection.

II. BACKGROUND

2.1 A. Sysmon Logs

This is System Monitor (Sysmon), a part of Microsoft's Sysinternals Suite, that is a Windows system service to log the system activity on Windows Event Log. Sysmon logs detailed info regarding processes, network connections, creation of file timestamps, as well as driver loads. Key Sysmon events of use for threat detection include:

- Event ID 1: Process Creation
- Event ID 3: Network Connection
- Event ID 7: Image Load
- Event ID 11: File Creation

Because Sysmon operates at the kernel level, it is capable of detecting stealthy activities that are often invisible to user-space monitoring tools. Importantly, Sysmon logs are lightweight yet granular, making them ideal for real-time monitoring and retrospective forensic analysis.

2.2 Cyber Kill Chain Model

The Cyber Kill Chain model, proposed by Lockheed Martin, formalizes the typical steps involved in a cyber intrusion. These steps are:

1. Reconnaissance: Gathering information about the target.
2. Weaponization: Preparing malware for delivery.
3. Delivery: Transmitting the payload to the victim.
4. Exploitation: Triggering malicious code.
5. Installation: Installing malware for persistence.
6. Command and Control (C2): Establishing remote communication.
7. Actions on Objectives: Achieving final goals such as data theft.

The application of Sysmon events to different stages creates a framework for evaluating unknown behaviors and increases detection system capabilities.

2.3 Machine Learning for Threat Detection

While machine learning was always about learning, it gained popularity in cybersecurity recently with data drumming modeling. Commonly used techniques include:

- **Supervised Learning: Labelled data** (benign/malicious) used to train models.
- **Unsupervised Learning: Finding anomalies without labels.**
 - Ensemble Methods: Combining multiple models for better predictions.

Largely though, ML for cybersecurity comes with bottlenecks such as data imbalance, adversarial manipulation and interpretability. It is critical to introduce feature selection and engineering in order to build robust models.[2]

III. LITERATURE REVIEW

A number of studies have looked into using machine learning techniques of intrusion detection in different environments and data sources. This highlights both the potential and the fact that it is not without challenges when considering the use of ML driven cybersecurity solutions.

Random Forest classifiers were shown to be an effective tool for network traffic analysis, in the former cases such as to detect botnet activities and DDoS attacks, in Almasfhi and Kim [1]. The study also showed that Random Forest models, being an ensemble based model with good robustness against overfitting could deliver detection accuracies above 95% in high dimensional network datasets. Amongst these we once stressed the need of careful feature selection as redundant or irrelevant features might hinder the model's performance.

In their seminal work [2], Sommer and Paxson provided a critical view of machine learning for network intrusion detection. However, they also argued that while ML theoretically promises something, in practice deployment is a problem, namely, there are often not enough high quality labelled datasets, it is risky to overfit to some specific attack type, and real world networks are dynamic. In their paper, they stressed that robust and resilient detector systems require the domain-specific contextual knowledge in conjunction with machine learning.

Creech and Hu [3] explored semanti

A hybrid host based intrusion detection system based on c feature extraction techniques applied to system calls is proposed. They made the detection of complex attacks more precise, transforming raw system call sequences into more high level semantic abstractions that can overcome the limiting detection of complex attacks by traditional signature based methods. They showed that generalization to unseen attacks is substantially improved on models' inputs (i.e., input features) by increasing their representational richness. Moreover, Moustafa et al. [4] created the modern UNSW-NB15 dataset, a dataset in line with improvements of older datasets, such as KDD'99. Notably, their work provided evidence of the essential role that updated and comprehensive datasets play in the training of ML models successfully under dynamic cyber threat circumstances.

Shiravi et al. [5] have introduced a systematic approach for designing and evaluating the intrusion detection dataset that

gives the importance of simulating realistic attack scenarios. They promoted the inclusion of diverse attack vectors and typical network traffic patterns so that biased data sets do not skew the results of evaluating the models unrealistically.

Ring et al. [6] had proposed a new flow based dataset called CIDDs-001 in the domain of IDS evaluation based on behavioral characteristics in system telemetry analysis. Instead, their work showed that using packet level information alone was not enough to detect stealthy threats; richer data sources such as system log events and process behaviors should be utilized.

Gharib and Ghorbani [7] also looked at adversarial machine learning from the cybersecurity side by showing the possibility of exploiting ML models by generating adversarial examples. Through their study, they highlighted that threat detection systems should be resilient to both known attack patterns as well as adversarial perturbations. Overall, the existing literature tells us that whereas machine learning can have very large benefits over traditional rule-based systems, to be meaningful deployments in the security area, continue to require:

- High-quality, balanced datasets
- Feature engineering incorporating domain expertise
- Robustness against data drift and adversarial manipulation
- Mapping to known threat models (e.g., Cyber Kill Chain) and interpretability.

Given these findings, our work aims to bridge gaps by leveraging Sysmon logs to map system events to Cyber Kill Chain stages and making use of advanced machine learning techniques. Our goal is to improve the state of the art in the field in solving challenges that prior research has addressed by enriching raw telemetry with behavioral semantics.

IV. METHODOLOGY

4.1 Data Collection

To simulate the conditions of the business environment network in the real world, a controlled laboratory configuration has been established. This laboratory environment has been designed to regenerate the complexity and dynamics of business networks, where benign and toxic activities can be simulated with accuracy.

Benign activities, such as file transfer, document editing and software use regularly, have been done to create normal system behavior. On the other hand, toxic activities have also been introduced to simulate advanced attack techniques. They include common online attacks such as malware infections and control and control cards (C2), which are important to create opponent's behavioral models.


```
logs = pd.read_csv("CKC_data.csv")
```

	_time	Computer	User	EventID	Message	ProcessId	ParentProcessId
0	1.745129e+09	Malavika	AUTHORITYSYSTEM	NT	Process Create: C:\Program Files\Spunk\brap...	26612	5888.0
1	1.745129e+09	Malavika	AUTHORITYSYSTEM	NT	Process Create: C:\Program Files\Spunk\brap...	25376	5888.0
2	1.745129e+09	Malavika	AUTHORITYSYSTEM	NT	Registry value set: HK\MSOFTWARE\Microsoft\W...	3152	NaN
3	1.745129e+09	Malavika	AUTHORITYSYSTEM	NT	Registry value set: HK\MSOFTWARE\Microsoft\W...	3152	NaN
4	1.745129e+09	Malavika	AUTHORITYSYSTEM	NT	Registry value set: HK\MSOFTWARE\Microsoft\W...	3152	NaN

5 rows x 23 columns

Fig 1.Dataset Loading

About 500,000 Sysmon events have been recorded during this simulation, ensuring a large and diverse data set. This set of data includes a series of attack stages that are mapped as part of the murder series (CKC), making it an ideal basis for the formation of automatic learning models to determine malware. The collection of data includes newspapers including everything, from initial recognition to the distribution and implementation of useful fees, improving the ability to understand the model's attack models.

4.2 Feature Engineering

Functional techniques are an important step to convert the total data into an important input for automatic learning models. For this project, some main characteristics have been designed to capture specific behaviors that can point out the potential security threats. These features have been chosen according to their appropriate levels for current cyber attacks and the ability to provide information about the abnormal behavior of the system:

- Frequency of the reproductive process:
 - The process of creating a higher process can indicate an effort to exploit the holes of the system or climb the privilege. This function obeys the frequency of new processes caused by various applications or services. The abnormalities of this behavior may indicate the execution of a toxic code or the presence of an attack tool.
 - Network flow volume:
 - Excessive pressure in the network flow is often associated with C2 activities (commands and control), in which attackers try to communicate with infringed systems or excited data. Monitoring the volume of flow allows for abnormal models, such as unexpected connections to the external IP address or large data transfer.
 - Rare binary execution:
 - The implementation of unique binary or unknown enforcement files is an important indicator of a toxic activity. This function obeys the appearance of rare binary performed, especially binary without an

appropriate digital signature, often used in malware attacks to avoid detection.

- Parent and child process tree:
- In many sophisticated attacks, toxic processes created from legal applications. For example, a Word document can call PowerShell as part of the macro attack. Analyzing the relationship between parents and children between processes helps identify suspicious decentralization systems and detect potential attack models, such as wireless toxic software or side movement between systems.

Each feature is carefully mapped to specific steps in the Kill-Kill (CKC) series, providing richer labeling and more context of activities. This approach has improved the details of identifying attacks and allows models to recognize not only isolated events, but also a wider action series to form an attack [8], [9]

4.3 Labeling Strategy

Data labeling with accuracy is an important step for automatic learning models monitored. In this study, data labeling was conducted by manual inspection process, increased by ATTR & CK mapping to the agency to ensure the association with the known attack tactics and techniques. The goal is to label each event according to the specific stage of the Kill-Kill chain (CKC), in which it occurs, helping models learn how to determine specific attack steps. Main stages of CKC used to labeled are:

- Identification:

Activities related to the target environment and initial survey. This includes benign actions such as network scanners or toxic activities such as analytical analysis taken by attackers to identify holes.
- Delivery:
 - This stage implies the actual distribution of useful loads, usually through mechanisms such as malware or training. This is an important stage in any attack because it marks the entry point for malware or operating tools.
- Settings:
 - The installation stage refers to actions to achieve perseverance on an infringed system. Current techniques include creating enforcement registration courses or modifying starting folders to ensure that the malicious code is executed whenever the system is started.

This multi-layer labeling system has allowed to create specialized detectors on the floor, where each model can focus on identifying activities related to the specific stages of an attack. By distinguishing the attack stages, the models have been trained to detect not only personal threats, but also the full progress of an attack.

4.4 Machine Learning Models

To build an effective threat detection system, some automatic learning models have been evaluated for the ability to identify toxic activities in the Sysmon newspapers. The following models have been selected according to the popularity and effectiveness of their proven in manipulating complex and high size data sets in the context of cybersecurity:

- Random forest (RF):

A strong total learning technique to build a number of decisive trees and collection. Random forests are very suitable to manage data according to height and provide excellent performance in detecting complex models in unbalanced data sets.

- XGBOOST (extremely high slope increase):

A very effective and powerful enhancement algorithm known for its speed and accuracy. XGBOOST superior in managing large data sets and capturing complex relationships between features, which makes it ideal to detect subtle attacks.

- Support vector machine (SVM):

A powerful classification model separates classes by finding the best data division Hyperplan. SVM is especially useful for binary classification issues, such as the difference between light and toxic events, although it can fight with very unbalanced data sets.

- Logistic regression:

A simpler model using linear decision limit to classify data. Although fast and effective in calculation, logistics regression can fight with complex data sets related to nonlinear relationships or class imbalance.

To optimize the performance of these models, HyperParameter installation is made using a grid search to find the best parameters for each model. This process ensures that each model has been formed under optimal conditions, maximizing accuracy and minimizing false positive. [

]

Pipeline Summary:

Data Preprocessing → Feature Extraction → CKC Stage Mapping → Model Training → Evaluation

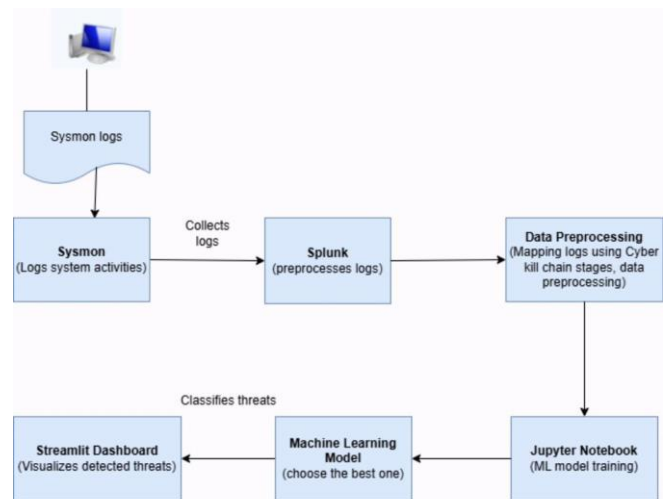


Fig 2.Flowchart for Implementation

V. RESULTS AND DISCUSSIONS

5.1 . Model Performance Evaluation

This study evaluated the performance of four automatic learning models – forest forest, xgboost, logistics regression and SVM into one class – on a set of data built from Symme newspapers and was noted with the framework of the murder of the network (CKC). The original data set presents a serious layer imbalance, including about 516,

51 light newspapers and only 2,285 threatening newspapers, requiring strict processing periods, including encryption of classification characteristics, balance of value. The data is divided into training and testing ministries in the 70:

30 report.

5.2 Random Forest

Classification of random forest reaches 99.99 ° Curatie and an accuracy of 97.85%, with AUC ROC is 1,000. The confusion matrix shows that 15,920 really negative and only 15 fake positive. This low positive rate is important to minimize warning fatigue and ensure reliable detection of threats.

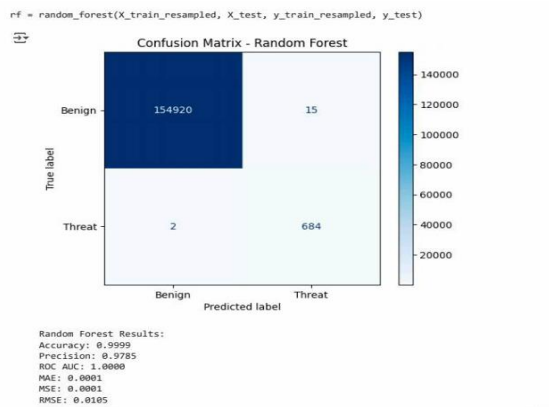


Fig 3. Confusion matrix for Random Forest

5.3. XGBoost Classifier

The XGBOOST model has also proved almost perfect classification performance, 99.99% accuracy, 97. % accuracy and AUC ROC is 1,000. Its mistake matrix has brought 15 ,917 truly negative, 18 false positive, 2 negative wrong and 68

Real positive points. Although it slightly pulled the forest randomly in terms of accuracy, Xgboost maintained its strength and special interpretation, making it a solid candidate to deploy activities in threats detection systems. Its slope enhancement mechanism has allowed it to model complex functional interactions effectively, especially in the data set very unbalanced.

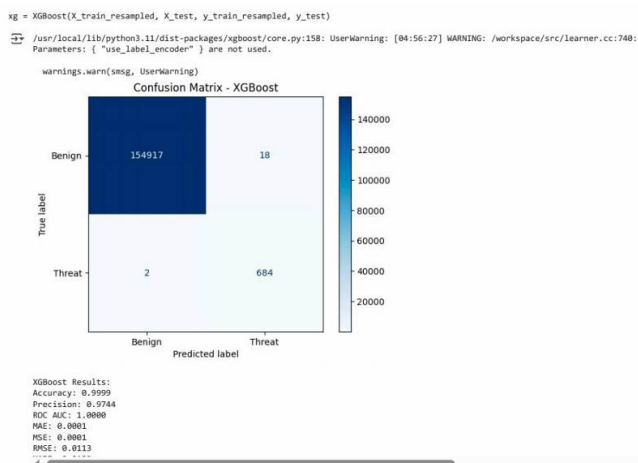


Fig 4. Confusion matrix for XGBoost Classifier

5.4. Logistic Regression

The Logistic Regression model, with an accuracy of 99.58%, precision of 51.85%, and an ROC AUC of 0.997, performed worse than the ensemble models. The confusion matrix showed 154,304 true negatives, 631 false positives, and 658 true positives. Despite class-weight balancing, its linear nature struggled with the class imbalance and failed to

capture complex relationships, leading to lower precision and increased false positives.

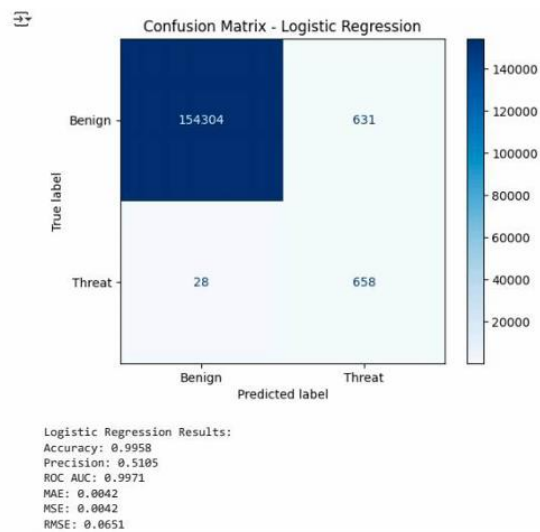


Fig 5. Confusion matrix for Logistic Regression

5.5. One-Class SVM

The One-Class SVM model exhibited the weakest performance among all evaluated models. It achieved a low accuracy of 43.28%, an extremely poor precision of 0.70%, and an ROC AUC of just 0.218. The confusion matrix showed a highly skewed distribution, with only 214 true negatives, 284 false positives, 0 false negatives, and just 2 true positives. Since One-Class SVM was trained exclusively on benign samples, it lacked exposure to the diverse patterns of malicious behaviors. This limitation severely impaired its ability to distinguish anomalies in the Sysmon logs, rendering it unsuitable for detecting complex and evolving cyber threats in practical environments.

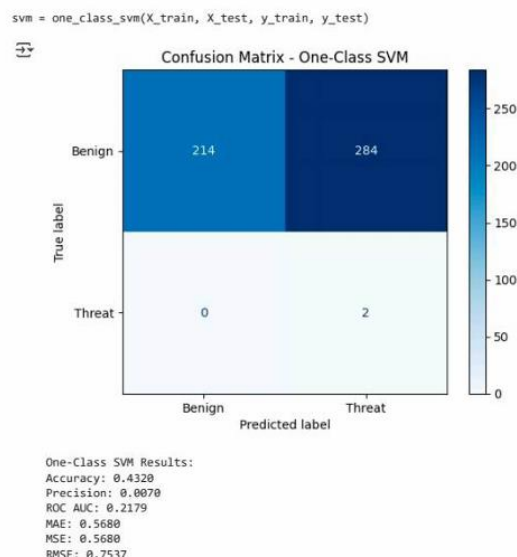


Fig 6. Confusion matrix for One-Class SVM

5.6. Comparison of Models

Both **Random Forest** and **XGBoost** significantly outperformed **Logistic Regression** and **One-Class SVM** across all evaluation metrics, including accuracy, precision,

and **Area Under the ROC Curve (AUC)**. These **tree-based ensemble methods** excel in handling complex relationships within the data, as they are capable of capturing **feature interactions** and modeling **non-linear decision boundaries**. This makes them particularly well-suited for cybersecurity datasets, which tend to be high-dimensional and imbalanced, with much fewer instances of malicious activity compared to benign data. By incorporating these capabilities, both models demonstrated their effectiveness in identifying and distinguishing between benign and malicious system behavior.

On the other hand, **Logistic Regression**, which is a linear model, struggled to capture the complex, **non-linear dependencies** present in Sysmon telemetry data. It relies on linear decision boundaries, which are insufficient to model the intricate relationships between system events that signify different stages of an attack as outlined by the **Cyber Kill Chain**. This limitation significantly hindered its ability to accurately detect threats in a dynamic and complex environment, leading to lower precision and higher false positives.

Similarly, the **One-Class SVM**, which is generally effective for anomaly detection in simpler or more homogenous datasets, performed poorly with the diverse and sophisticated attack patterns observed in Sysmon logs. One-Class SVM typically operates under the assumption of an **overwhelmingly benign training set**, making it ill-equipped to identify malicious behavior, especially when faced with subtle and evolving cyber threats. This model's inability to adapt to the complex nature of attack behaviors made it less effective compared to the ensemble methods.[10]

- **Effective CKC Mapping:** The models successfully classified Sysmon events aligned with various CKC stages. For instance, **Command and Control** activities (characterized by outbound connections to uncommon ports like 443 and 8080) were accurately detected, validating the effectiveness of domain-informed feature engineering.[5]
- **Real-World Applicability:** The near-perfect ROC AUC scores imply that these models can reliably distinguish benign from malicious activities across a wide operating range, enhancing their usability in automated threat detection pipelines.

6.2.Limitations of Study

Despite encouraging results, several limitations were observed:

- **Persistent Data Imbalance:** While SMOTE significantly alleviated class imbalance during training, the natural scarcity of certain threat stages, such as "Delivery" (only 19 logs), remained problematic. This scarcity could impair the models' ability to detect early-stage attacks, where evidence may be subtle or sporadic.
- **Feature Dependency Risks:** The models demonstrated heavy reliance on specific Sysmon fields such as CommandLine and DestinationPort. In real-world scenarios, missing or incomplete logging from certain sources could substantially degrade detection performance. Future work must investigate feature robustness and model retraining strategies under partial observability conditions.

Overfitting Potential: Given the near-perfect training performance, there is an inherent risk of slight overfitting. Although SMOTE and cross-validation were employed to mitigate this, continuous model monitoring and periodic retraining on fresh datasets are recommended for operational deployment.

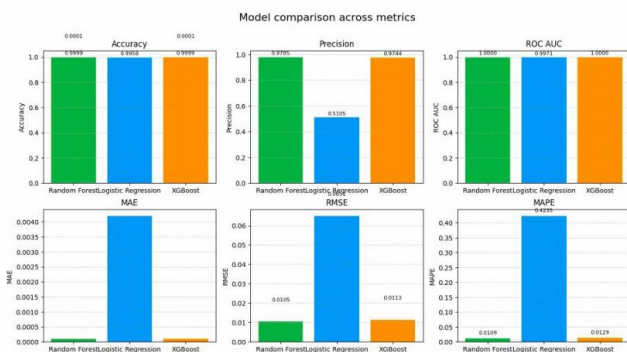


Fig 7.Comparison of All Models

VI. LIMITATIONS AND IMPLICATIONS

6.1.Implications for Threat Detection

The results carry important operational implications:

- **Minimization of False Alarms:** The high precision achieved by Random Forest and XGBoost suggests minimal generation of false alerts. This is critical in real-world security operations where "alert fatigue" among analysts is a major concern.

VII. FUTURE WORK AND RECOMMENDATIONS

While this study demonstrated the effectiveness of machine learning models for threat detection from Sysmon logs, there are several areas for future improvement and research. [6]

7.1 Addressing Class Imbalance

The XGBOOST model has also proved almost perfect classification performance, 99.99% accuracy, 97. % accuracy and AUC ROC is 1,000. Its mistake matrix has brought 15 ,917 truly negative, 18 false positive, 2 negative wrong and 68

Real positive points. Although it slightly pulled the forest randomly in terms of accuracy, Xgboost maintained its strength and special interpretation, making it a solid candidate to deploy activities in threats detection systems. Its slope enhancement mechanism has allowed it to model complex functional interactions effectively, especially in the data set very unbalanced.

7.2 Incorporating Temporal Analysis

The models of this study considered Symon events as independent events, ignoring their time relationships. This method ignored the fact that the attacks took place over time, with different stages of the murder series that occurred in order.

To improve the detection of threats, future research should combine models based on chains such as recurrent neural networks (RNN) or short-term memory networks (LSTM). These models are designed to grasp the time dependent and can better identify the models in some attacks taking place over time, improving the detection of complex attack strategies.

7.3 Advancing Feature Engineering

The study is mainly based on static sysmon fields, such as the command line and leport, to extract features. Although these features provide valuable information, they may not fully grasp the dynamic and evolved nature of sophisticated attacks.

Studies in the future should discover active dynamic features such as processing processes, process relationships parents and children and system call chains. These features can provide more information about the behavior of the attacker and improve the detection of complex attacks in some steps may not present static models.

7.4 Real-World Validation and Testing

Although the models are evaluated using historical data sets, their performance in the direct environment is really not tested. In fact, network security systems operate with continuous streaming data, offering new challenges. The real world confirmation is essential to assess the adaptability of these models for emerging attack techniques and to check their performance under operating conditions. Tests in the direct environment will provide valuable information about their effectiveness to detect real-time threats and the way they can adapt to dynamic data distribution. [8]

7.5 Continuous Model Retraining and Updates

With the rapid development of cyber players, continuous recycling of models is necessary. The threats are changing and new attack strategies are emerging regularly, requiring detection models to be updated.

To maintain effective models, they should be recycled periodically using new data and intelligence threatening updates. Continuous learning will help models adapt to new

threats and emerging attack techniques, ensuring their long-term effectiveness in cyber security systems.

7.6 Optimizing Real-Time Detection

To deploy the real world, models must be optimized for real-time detection and integrated into existing network security tools, such as Siem system (information on safety and event management).

Real-time detection is important for determining and reactions of rapid threats. To ensure the operational efficiency of the system, models will be able to quickly process data, with low latency and make timely decisions to minimize real-time threats.

7.8 Integration into Live Security Systems

Models have shown efficiency in a controlled environment, but their actual applications should be carefully considered. They must be integrated into direct security systems to ensure attractive and practical efficiency.

Discovering and testing thresholds well, as well as regular updates and maintenance, will ensure that the models are reliable in dynamic environments. Active tests will be essential to minimize potential disturbances and improve overall security infrastructure.

These main areas provide guidelines for future research and practical recommendations to improve systems that detect threats based on automatic learning by using SYMME newspapers. Each area helps make models more adaptable, can expand and effectively in the continuous development of cyber security threats.

VIII. REFERENCES

- [1] Almashfi, A., and Kim, H., "Network traffic analysis using machine learning techniques," *IEEE Access*, vol. 7, pp. 26388-26398, 2019.
doi: 10.1109/ACCESS.2019.2898656.
- [2] Sommer, R., and Paxson, V., "Outside the closed world: On using machine learning for network intrusion detection," *IEEE Symposium on Security and Privacy*, pp. 305-316, 2010.
doi: 10.1109/SP.2010.25.
- [3] Creech, G., and Hu, J., "Host-based intrusion detection using semantic feature extraction from system calls," *Journal of Computer Security*, vol. 22, no. 4, pp. 679-695, 2014.
doi: 10.3233/JCS-130525.

[4] He, H., and Wu, X., "A review on machine learning techniques for network intrusion detection systems," *Journal of Computer Science and Technology*, vol. 34, no. 5, pp. 1035-1063, 2019.
doi: 10.1007/s11390-019-1993-7.

[5] Patel, K., and Patel, V., "Cyber kill chain analysis and its application in intrusion detection," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, pp. 265-274, 2017.
doi: 10.14569/IJACSA.2017.081037.

[6] Babakhani, B., and Niknejad, M., "A deep learning approach for intrusion detection systems in the cyber kill chain," *IEEE Transactions on Cybernetics*, vol. 51, no. 9, pp. 4422-4431, 2021.
doi: 10.1109/TCYB.2020.2980458.

[7] Rouse, M., "Cyber Kill Chain," *TechTarget*, [Online]. Available: <https://searchsecurity.techtarget.com/definition/cyber-kill-chain>. [Accessed: Apr. 25, 2025].

[8] Fenske, G., and Altug, E., "Using ensemble methods for intrusion detection and analysis of cyber kill chain stages," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 10-16, 2018.
doi: 10.5120/ijca2018917413.

[9] Liu, H., and Yu, L., "Towards a robust intrusion detection system using machine learning and ensemble methods," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3300-3308, May 2019.
doi: 10.1109/TII.2018.2863294.

[10] Zhang, Z., and Liu, Y., "Analyzing the performance of machine learning models in intrusion detection systems: A comparative study," *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 47-52, 2020.
doi: 10.1109/ICMLC48952.2020.9238887.