A User-Centric Machine Learning Framework for Cyber Security Operations Center

Charles Feng*
ZhongDu Technologies, Inc.
Shaoxing, Zhejiang, China
CharlesFeng99@gmail.com
*Corresponding Author

Shuning Wu
Center for Advanced Machine Learning
Symantec Corporation
Mountain View, California, USA
Shuning_Wu@Symantec.com

Ningwei Liu
Norton Business Unit
Symantec Corporation
Mountain View, California, USA
Ningwei_liu@Symantec.com

***Abstract*—To assure cyber security of an enterprise, typically SIEM (Security Information and Event Management) system is in place to normalize security events from different preventive technologies and flag alerts. Analysts in the security operation center (SOC) investigate the alerts to decide if it is truly malicious or not. However, generally the number of alerts is overwhelming with majority of them being false positive and exceeding the SOC's capacity to handle all alerts. Because of this, potential malicious attacks and compromised hosts may be missed. Machine learning is a viable approach to reduce the false positive rate and improve the productivity of SOC analysts. In this paper, we develop a user-centric machine learning framework for the cyber security operation center in real enterprise environment. We discuss the typical data sources in SOC, their work flow, and how to leverage and process these data sets to build an effective machine learning system. The paper is targeted towards two groups of readers. The first group is data scientists or machine learning researchers who do not have cyber security domain knowledge but want to build machine learning systems for security operations center. The second group of audiences are those cyber security practitioners who have deep knowledge and expertise in cyber security, but do not have machine learning experiences and wish to build one by themselves. Throughout the paper, we use the system we built in the Symantec SOC production environment as an example to demonstrate the complete steps from data collection, label creation, feature engineering, machine learning algorithm selection, model performance evaluations, to risk score generation.**

Keywords—user-centric; machine learning system; cyber security operation center; risky user detection

I. INTRODUCTION

Cyber security incidents will cause significant financial and reputation impacts on enterprise. In order to detect malicious activities, the SIEM (Security Information and Event Management) system is built in companies or government. The system correlates event logs from endpoint, firewalls, IDS/IPS (Intrusion Detection/Prevention System), DLP (Data Loss Protection), DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol), Windows/Unix security events, VPN logs etc. The security events can be grouped into different categories [1]. The logs have terabytes of data each day.

From the security event logs, SOC (Security Operation Center) team develops so-called use cases with a pre-determined severity based on the analysts' experiences. They are typically rule based correlating one or more indicators from different logs. These rules can be network/host based or time/frequency based.

If any pre-defined use case is triggered, SIEM system will generate an alert in real time. SOC analysts will then investigate the alerts to decide whether the user related to the alert is risky (a true positive) or not (false positive). If they find the alerts to be suspicious from the analysis, SOC analysts will create OTRS (Open Source Ticket Request System) tickets. After initial investigation, certain OTRS tickets will be escalated to tier 2 investigation system (e.g., Co3 System) as severe security incidents for further investigation and remediation by Incident Response Team.

However, SIEM typically generates a lot of the alerts, but with a very high false positive rate. The number of alerts per day can be hundreds of thousands, much more than the capacity for the SOC to investigate all of them. Because of this, SOC may choose to investigate only the alerts with high severity or suppress the same type of alerts. This could potentially miss some severe attacks. Consequently, a more intelligent and automatic system is required to identify risky users.

The machine learning system sits in the middle of SOC work flow, incorporates different event logs, SIEM alerts and SOC analysis results and generates comprehensive user risk score for security operation center. Instead of directly digging into large amount of SIEM alerts and trying to find needle in a haystack, SOC analysts can use the risk scores from machine learning system to prioritize their investigations, starting from the users with highest risks. This will greatly improve their efficiency,

978-1-5090-6727-5/17/$31.00 ©2017 IEEE

optimize their job queue management, and ultimately enhance the enterprise's security.

Specifically, our approach constructs a framework of user-centric machine learning system to evaluate user risk based on alert information. This approach can provide security analyst a comprehensive risk score of a user and security analyst can focus on those users with high risk scores.

To the best of our knowledge, there is no previous research on building a complete systematic solution for this application. The main contribution of this paper is as follows:

- An advanced user-centric machine learning system is proposed and evaluated by real industry data to evaluate user risks. The system can effectively reduce the resources to analyze alerts manually while at the same time enhance enterprise security.
- A novel data engineering process is offered which integrates alert information, security logs, and SOC analysts' investigation notes to generate features and propagate labels for machine learning models.

II. RAW DATA AND DATA PREPROCESSING

The raw data is collected from Symantec internal security logs. It consists of alerts from SIEM system, notes from analysts' investigation, and logs from different sources, including firewall, IDS/IPS, HTTP/FTP/DNS traffic, DHCP, vulnerability scanning, Windows security event, VPN and so on.

For network traffic data or other data sources with dynamic IP addresses, entity resolution is required as in large enterprises, many internal IP addresses are dynamically assigned to users, causing the IP addresses of users to change over time. Without accurate dynamic IP to user mapping, the correlation of activities over different network logs will be challenging and inaccurate. IP to user mapping is applied to network traffic data to make sure that user ID is appended as primary key for data summarization, data correlation and feature engineering.

Finally, analyst's investigation notes are usually stored in a ticketing system as free-form text. The notes typically include the following information: the reason why an alert was triggered, supporting information from internal system logs and external resources (such as VirusTotal and IPVoid), and investigation conclusion on whether an alert is true positive or not.

III. USER FEATURE ENGINEERING AND LABEL GENERATION

A. Feature Creation

The features are created at individual user level as our main goal is to predict the user's risk. We have created over 100 features to describe a user's behavior. The features include: summary features created from statistical summaries (number of alerts per day), temporal features generated from time series analysis (event arrival rate), relational features derived from social graph analysis (user centrality from user-event graph) , etc.

B. Label Generation and Propagation

After all features are generated, we need to attach target or "label" for our machine learning models. The initial labels are created by mining analyst's investigation notes. Text mining techniques, such as key word/topic extraction and sentiment analysis, are used to extract the user's actual state from the notes.

From the users with annotations, generally very few of them (<2%) will be marked as "risky" after text mining. There are two concerns if we only use these users for machine learning:

- Majority of the users without annotations are left out of model, but they may have valuable information
- Many machine learning models do not work well for highly unbalanced classification problem

In order to alleviate these two issues, label propagation techniques are needed to derive more labels. The main idea here is, if we have knowledge about certain risky users, we can label other users with "similar" behaviors as risky. The label propagation techniques we used include: Matrix factorization-based clustering and Supervised PU learning [2].

Finally, we combine the labels from text mining and label propagation as the targets for our machine learning models. The final analytic dataset will look like this:

TABLE I. EXAMPLE ON FINAL MODELING DATASET

User ID	Summary feature 1	Indicator feature 2	Temporal feature 3	Relational feature 4	…	Label
User1	13	1	0.65	5.17	…	1 (risky - Initial)
User2	25	0	2.74	9.34	…	1 (risky - Derived)
User3	4	0	1.33	3.52	…	0 (normal)

IV. MACHINE LEARNING ALGORITHMS AND IMPLEMENTATIONS

A. Machine Learning Algorithms

In our system, we tried several machine learning algorithms [3][4][5][6][7], including Multi-layer Neural Network (MNN) with two hidden layers, Random Forest (RF) with 100 Gini-split trees, Support Vector Machine (SVM) with radial basis function kernel and Logistic Regression (LR). In our practice, we find that Multi-layer Neural Network and Random Forest work pretty well for our problem. Some validation results from these models will be shown later.

B. Model Performance Measures

As a common practice, modeling data should be randomly split into training and testing sets and different models should be evaluated on test holdout data. Besides AUC, we also define two measures of model goodness in Equations (1) to (2) below:

$$\text{Model Detection Rate} = \frac{Number\ of\ Risky\ Hosts\ in\ Certain\ Predictions}{Total\ Number\ of\ Risky\ Hosts} \times 100\% \quad (1)$$

$$\text{Model Lift} = \frac{Proportion\ of\ Risky\ Hosts\ in\ Certain\ Predictions}{Overall\ Proportion\ of\ Risky\ Hosts} \quad (2)$$

In contrast to the AUC that evaluates model on the whole test data, detection rate and lift reflect how good the model is in discovering risky users among different portions of predictions. To calculate these two metrics, the results are first sorted by the

model scores (in our case, the probability of a user being risky) in descending order. Detection rate measures the effectiveness of a classification model as the ratio between the results obtained with and without the model. For example, suppose there are 60 risky users in test data, from top 10% of the predictions, the model captures 30 risky users, the detection rate is equal to 30⁄60=50%. Lift measures how many times it is better to use a model in contrast to not using a model. Using the same example above, if the test data has 5,000 users, the lift is equal to (30/500)/(60/5000)=5. Higher lift implies better performance from a model on certain predictions.

C. Model Validation Results

To validate the effectiveness of our machine learning system, we take one month of model running results and calculate the performance measures. We split the data randomly into training (75% of the samples) and testing (remaining 25%) sets. The table below lists different models' AUC on test data. With average AUC value over **0.80**, Multi-layer Neural Network and Random Forest achieves satisfying accuracy.

TABLE II. MODEL AUC ON TEST DATA

	MNN	RF	SVM	LR
MEAN	**0.807**	**0.829**	0.775	0.754
STANDARD ERROR	0.006	0.004	0.016	0.008

Table III lists the detection rates for different models on top 5% to 20% predictions respectively. It is promising that Random Forest is able to detect 80% of the true risky cases with only 20% highest predictions.

TABLE III. MODEL DETECTION RATES ON TOP 5%~20% PREDICTIONS

Top % Predictions	MNN	RF	SVM	LR
5%	31.67%	25.00%	20.00%	31.67%
10%	58.33%	43.33%	46.67%	50.00%
15%	70.00%	70.00%	70.00%	68.33%
20%	78.33%	**80.00%**	80.00%	76.67%

Finally, we evaluate the model lift also on top 5% to 20% predictions as listed in Table IV. For top 5% predictions, Multi-layer Neural Network achieves lift value of 6.82, meaning that it is almost **7 times** better than current rule-based system. If we look at the average lifts on top 5% to 20% predictions, Multi-layer Neural Network is the highest with average lift over 5.5 as listed on the last row of Table V. This is very encouraging.

TABLE IV. MODEL LIFTS ON TOP 5%~20% PREDICTIONS

Top % of Predictions	MNN	RF	SVM	LR
5%	**6.82**	5.30	4.19	6.82
10%	6.25	4.55	4.92	5.30
15%	4.92	4.92	4.92	4.80
20%	4.09	4.19	4.19	4.00
Average	***5.52***	*4.74*	*4.56*	*5.23*

D. Model Implementations and Active Learning

Currently the machine learning system has been implemented in a real enterprise production. The features and labels are being updated daily from historical data. Then the machine learning model is refreshed and deployed to the scoring engine daily to make sure it captures the latest patterns from the data. After that, the risk scores are generated in real time when new alerts are triggered, so SOC analysts can take action right away for high risk users. Finally, SOC analysts' notes will be collected and fed back into historical data for future model refinement. The whole process has been streamlined automatically from data integration to score generation. The system also actively learns new insights generated from analysts' investigations.

V. CONCLUSIONS AND DISCUSSIONS

In this paper, we present a user-centric machine learning system which leverages big data of various security logs, alert information, and analyst insights to the identification of risky user. This system provides a complete framework and solution to risky user detection for enterprise security operation center. We describe briefly how to generate labels from SOC investigation notes, to correlate IP, host, and users to generate user-centric features, to select machine learning algorithms and evaluate performances, as well as how to such a machine learning system in SOC production environment. We also demonstrate that the learning system is able to learn more insights from the data with highly unbalanced and limited labels, even with simple machine learning algorithms. The average lift on top 20% predictions for multi neural network model is over 5 times better than current rule-based system. The whole machine learning system is implemented in production environment and fully automated from data acquisition, daily model refreshing, to real time scoring, which greatly improve SOC analyst's efficiency and enhance enterprise risk detection and management. As to the future work, we will research other learning algorithms to further improve the detection accuracy.

REFERENCES

[1] SANS Technology Institute. "The 6 Categories of Critical Log Information." 2013.

[2] X. Li and B. Liu. "Learning to classify text using positive and unlabeled data", Proceedings of the 18th international joint conference on Artificial intelligence, 2003

[3] A. L. Buczak and E. Guven. "A survey of data mining and machine learning methods for cyber security intrusion detection", IEEE Communications Surveys & Tutorials 18.2 (2015): 1153-1176.

[4] S. Choudhury and A. Bhowal. "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection", Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015.

[5] N. Chand et al. "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection", Advances in Computing, Communication, & Automation (ICACCA), 2016.

[6] K. Goeschel. "Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis", SoutheastCon, 2016.

[7] M. J. Kang and J. W. Kang. "A novel intrusion detection method using deep neural network for in-vehicle network security", Vehicular Technology Conference, 2016.