A Machine Learning Approach for Network Traffic Analysis using Random Forest Regression

Shilpa Balan
College of Business and Economics, Department of Information Systems
California State University, Los Angeles
sbalan@calstatela.edu

Pamella Howell
College of Business and Economics, Department of Information Systems
California State University, Los Angeles
phowell@calstatela.edu

Abstract

The Internet is a necessary part of our daily lives. Although the Internet has many benefits, it can compromise the security of the systems connecting to it in numerous ways. Resultantly, attacks on networks have increased in number and severity over the past few years; hence, Intrusion Detection Systems (IDSs) are a significant part of an organizations' infrastructure. Intrusion detection systems help reduce security risks by improving the network ability to resist external attacks. The objective of this paper is to examine the features impacting Brute Force SSH and FTP attacks using the Random Forest machine learning technique. We utilize a data set that includes updated network attacks and simulates real-world traffic flow. Using realistic traffic features, our prediction model achieved high accuracy when identifying Brute Force SSH and FTP attacks.

Introduction

The Internet is a necessary part of our daily lives. The Internet is useful in several areas, such as business, entertainment, education, among others. In particular, the Internet is an essential component of business models [1]. Although the Internet has many benefits, it can also compromise the security of the systems connecting to it in numerous ways. Firewalls are a vital part of network security. However, more dynamic mechanisms such as intrusion detection systems (IDSs) should be utilized [2] due to the increasing sophistication of attacks on networks. “Intrusion detection is the process of monitoring events occurring in a computer system or network and analyzing them for signs of intrusions” [3, p.5].

Intrusion detection is an important research area for both business and personal networks [4]. A network attack occurs when a hacker maliciously attempts to compromise the security of a network. There are various types of attacks, categorized by the types of code and tools required to execute them. Brute Force attacks are among the most common perpetrated by hackers. The typical reasons for network attacks are financial gain, to damage and corrupt data, to steal data, to prevent legitimate authorized users from accessing network services, and for several other reasons [4].

Risks are an inherent part of the internet environment. Hence, intrusion detection systems (IDSs) are required to aid the networks capability to resist external attacks. As network attacks have increased in frequency and severity over the past few years, intrusion detection systems (IDSs) are a necessity in an organization [5]. The goal of an IDS is to defend and confront malicious attacks on computer systems from the Internet; whereas the conventional firewall cannot perform this task [6].

There are two different detection techniques employed in IDS to search for attack patterns: misuse and anomaly. Misuse detection systems find known attack signatures in the monitored resources. Anomaly detection systems find attacks by detecting changes in the pattern of the behavior of the system [7]. The extant literature outlines several anomaly detection systems developed based on different machine learning (ML) techniques. For example, some studies apply a single ML technique, such as neural networks or support vector machines. On the other hand, some detection systems are developed based on hybrid or ensemble machine learning techniques. In particular, these techniques are used to recognize whether access to the Internet is regular or an attack [7].

The goal of this paper is to review the patterns of a network attack using a single machine learning perspective. Machine learning techniques can automatically generate rules used for computer network intrusion detection [8]. In particular, we examine Brute Force Secure Shell (SSH) and File Transfer Protocol (FTP) attacks which are among the most popular attack scenarios [9]. There is an urgent need to automate the intrusion detection system to differentiate intrusive from non-intrusive network traffic due to the growing number of data calls. In this paper, we illustrate the detection of network attack using a Random Forest (RF) model.

This paper is organized as follows. The background section provides an overview of machine learning techniques and briefly describes some related techniques for intrusion detection. The methodology section describes the technology and machine learning algorithm used in this study. The analysis and results expound on the findings of the

research. Finally, the conclusion and discussion for future research are detailed at the end of the paper.

Background

Researchers are working to resolve the issue of increasingly intrusive network activities. There is considerable research and development on attack detection strategies, but only limited research on testing these techniques against realistic data [10]. There are several types of attacks launched every hour of every day. For example, browser-based network attacks are executed by hackers who attempt to breach a machine through a web browser which is one of the most common ways people use the Internet. Attackers breach the website and infect it with malware. When a user visits the website, the infected site attempts to force malware onto their systems by exploiting vulnerabilities in their browsers [11].

In this paper, we examine the patterns of SSH and FTP attacks using machine learning. SSH attacks are of various types: SSH port scanning, SSH Brute-force attacks, and attacks using compromised SSH server. Attacks using a compromised server could be DoS attacks, phishing attacks, and email spamming [12]. FTP attacks may include spoof attack, brute force, bounce attack, packet capture, and port stealing [13, 14]. This paper examines whether the attacks from an SSH or FTP server could be segregated from other attacks using the network flows.

An IDS could be trained to recognize the client addresses that typically access a particular server by observing it over some training period [15]. Naïve Bayes, Nearest Neighbor, and Neural Networks are among the machine learning techniques previously applied to network attack detection [15, 8]. In an experiment applied to the KDD'99 data set, researchers used Random Forest (RF) regression for misuse detection [16]. Their analysis was conducted using the Weka software for data mining.

Previous researchers also evaluated the KDD'99 data set using Support Vector Machine (SVM) and the Random Forest (RF) algorithm [17]. The study results indicate that a Random Forest approach takes less time to train the classifier than SVM. Their paper further purports that continued research on intrusion detection using SVM and RF is viable due to their excellent performance.

Network intrusion detection systems must distinguish between hostile and benign traffic. In this paper, we apply Random Forest Regression using Python packages to examine the patterns of network attack flows from the SSH and FTP server.

Methodology

Data

This study used a data set generated by the Canadian Institute for Cybersecurity (CIC) and referred to as CICDS2017 [9]. A link to the data can be found at https://www.unb.ca/cic/datasets/ids-2017.html. The data set is particularly robust as it includes the finding of an updated evaluation framework [10]. This framework details the criterion necessary for building a reliable benchmark data set including in network configuration, traffic, protocols, diversity, and heterogeneity.

The CIC data capture started at 9:00 a.m., Monday, July 3, 2017, and ended at 5:00 p.m. on Friday, July 7, 2017, for a total of 5 days. Researchers initiated six attacks profiles, including Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS attacks. They were executed on Tuesday, Wednesday, Thursday and Friday morning and afternoon. Monday is the typical day and only includes benign traffic [18]. The data used in our analysis were collected on Tuesday when researchers executed Brute Force SSH and Brute Force FTP attacks. The number of observations that include the FTP attacks is 7,937, and the number of observations that include the SSH attacks is 11,794.

The data set simulates real-world data as it contains both benign and conventional attacks. It includes the results of the network traffic analysis using CICFlowMeter [19] with labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols, and attack. NetFlowMeter is a network traffic flow generator written in Java. CICFlowMeter generates Bidirectional Flows (Biflow), where the first packet determines the forward (source to destination) and backward (destination to source) directions. Hence, the features such as duration, number of packets, number of bytes, and length of packets are also calculated in the forward and reverse direction. The output of the application is the CSV file format with columns labeled for each flow, namely Flow ID, Source IP, Destination IP, Source Port, Destination Port, and Protocol with more than 80 network traffic features [19].

The features included in our analysis were extracted and validated by CIC researchers using Random Forest Regressor. The selected variables shown in Table 1 are considered the best detection features for each type of attack [9].

Table 1. Network Attacks and Features

Network Attack	Feature
SSH-Patator	Init Win F.Bytes
	Subflow F.Bytes
	Total Len F.Packets
	ACK Flag Count
FTP-Patator	Init Win F.Bytes
	F.PSH Flags
	SYN Flag Count
	F.Packets/s

As suggested by previous researchers [9], the features we used for analyzing the patterns of SSH attacks are Init Win F.Bytes (initial window of forward bytes), Subflow F.Bytes (subflow of forward bytes), Total Len F.Packets (total length of forward packets) and ACK Flag Count. The features we used for analyzing the patterns of attacks from an FTP port are Init Win F.Bytes, F.PSH Flags, SYN Flag count and F.Packets/s (forward packets per second). A description of the features is shown in Table 2 as suggested by CICFlowMeter [19].

Table 2. Description of Features

Feature Name	Description
Init Win F.Bytes	The total number of bytes sent in initial window in the forward direction
Subflow F.Bytes	The average number of bytes in a sub flow in the forward direction
Total Len F.Packets	Total length of packets in the forward direction
ACK Flag Count	ACK (Acknowledge) Flag count
F.PSH Flags	Number of times PSH (Push) flag was set in packets travelling in the forward direction
SYN Flag Count	Number of packets with SYN (Synchronization)
F.Packets/s	Total packets in the forward direction

Algorithm

We propose the Random Forest algorithm as an approach for intrusion detection in this study. Random Forest is an ensemble classification and regression approach [20]. The Random Forest algorithm has been used extensively in different applications. For instance, it has been applied to prediction [21, 22], probability estimation [23], and pattern analysis in multimedia information retrieval and bioinformatics [24]. Accuracy is a critical performance measure to develop an effective network intrusion detection system.

Figure 1 shows the methodology used in our paper. The data set is first extracted from the csv file format. From the extracted data set, the train and test data sets are built. Based on the features in the train data set, a model is built. The model is tested using the test data set; the test model is then used for predictions.

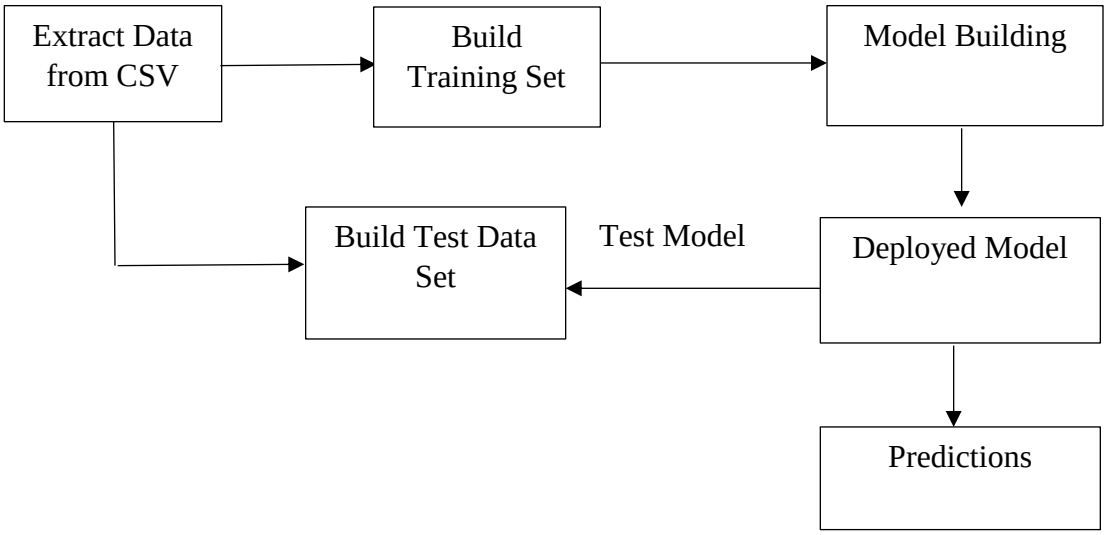

Figure 1: Architecture of Methodology

Results

In this paper, the performance and accuracy of the features were examined with the help of the Random Forest Regression machine learning algorithm. We validated patterns and accuracy of the FTP and SSH network flows and attacks using the features established by previous researchers [9] as shown in Table 1 in the Data section.

Pseudocode

The python scikit-learn packages were applied for machine learning prediction, as shown in Figure 2. Using Random Forest Regression, the factors impacting the SSH and the FTP attacks are independently replicated in this study.

We tested using the 80-20 percent split ratio and the 70-30 percent ratio. Our model accuracy with both split ratios provided perfect results, affirming the features established by previous researchers [9] for the CICDS2017 data set. The prediction accuracy resulted in 99.9% for both the types of attacks. The results of the analysis indicate that the prediction of SSH attack depends on the accuracy of the number of bytes in the initial window in the forward direction, the average number of bytes in a sub-flow in the forward direction, the total length of packets in the forward direction, and the acknowledgment flag count. The results also indicate that the FTP attack depends on the number of bytes in the initial window in the forward direction, the forward push flags, the synchronization flag count, and the number of packets in the forward direction. To verify this prediction accuracy of the data, other attack types can also be examined in the future.

```
Import packages
        from sklearn import metrics
        from sklearn.preprocessing import LabelEncoder
        from sklearn.metrics import accuracy_score
Read the security data file
Import train_test_split function
Split dataset into features and labels
        label_encoder = LabelEncoder()
        data.iloc[:,0] = label_encoder.fit_transform(data.iloc[:,0]).astype('float64')
adjust values for x and y (attributes and label)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
Import Random Forest Model
Create a Gaussian Classifier
        clf=RandomForestClassifier(n_estimators=100)
Train the model using the training sets y_pred=clf.predict(X_test)
        clf.fit(X_train, y_train)
        y_pred=clf.predict(X_test)
Evaluate the algorithm
Import scikit-learn metrics module for accuracy calculation
        accuracy=accuracy_score(y_test, y_pred)
```

Figure 2: Pseudocode of Machine Learning Algorithm

To evaluate the performance of the FTP and SSH Random Forest model, we used accuracy precision, recall and the F1 score. Precision or the positive predictive value (PPV) evaluates how many records were correctly returned [25]. Recall or true positive rate (TPR) measures the number of positive records the model returned [25]. The results of our analysis shows that precision was 0.96 and recall was 0.97. The F1 score is a weighted harmonic mean of precision and recall. We calculated the F1 score using the formula:

$$\text{F1 score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

For our study, F1 score was 0.965. As F1 score indicates the measure of the test's accuracy, the F1 score of 0.965 validates the accuracy of our model results.

Discussion

The features of a machine learning (ML) algorithm are crucial in determining the model's performance. In this section, we discuss the results with those of previous data sets.

The DARPA data set (created at the Lincoln laboratory in 1998-99) was constructed for network security analysis and exposed the issues associated with the artificial injection of attacks and benign traffic. The features of the DARPA data set include email, browsing, FTP, Telnet, IRC, and SNMP activities. It contains attacks such as DoS, guess password, buffer overflow, remote FTP, SYN Flood, Nmap, and rootkit. DARPA does not represent

real-world network traffic. The DARPA data set, unlike CICDS2017, is outdated for the practical evaluation of IDSs in terms of attack types. Therefore, it would be difficult for an organization to replicate its features during data capture, limiting the usefulness of ML algorithms for anomaly detection [26, 27].

The data set from the Lawrence National Laboratory and ICSI (2004-2005) is a full header network traffic. In an evaluation by other authors [27], they found that the ICSI data set does not have payload and suffered because the information which could identify individual IP addresses was removed to maintain anonymity.

The data set provided by the researchers [9] used for the current study is not anonymized; therefore, it is a realistic traffic monitoring data set that enables analysis for identifying the patterns of network flows. By using a labeled, updated data set like CICDS2017, the features of our prediction algorithm more accurately depict current attack types thus, reducing the impact of feature selection bias.

Conclusion and Future Research

The volume and variety of traffic on the Internet is increasing exponentially. In this paper, we presented a machine learning approach for network attack classification based on traffic behavior. By analyzing data collected for a short duration of traffic flow, the approach implemented in this study independently replicated the features of Brute Force SSH and FTP anomalies, providing a good understanding of the overall accuracy and improving the feature selection.

There are still a number of areas where future work is important. Further experiments could be carried out to extend performance evaluation and to demonstrate the ability to handle encrypted traffic and previously unknown applications, based on more traffic traces [29]. Moreover, using machine learning, it is easy to find similarities in comparison to outliers [30].

In future, it will be useful to find the outliers in the detection of the FTP and SSH attacks. While this paper explored the Brute Force FTP and SSH attacks using machine learning algorithm, the prediction accuracy and feature selection for other attacks can also be examined. Specifically, future studies should evaluate attacks implemented on other days of the week from the CICDS2017 data set to validate further the network features that determine the attacks. Moreover, researchers should strive to develop other data sets that simulate real-world network traffic or collect actual data to verify the results of our study.

References

[1] Shon, T., and Moon, J. "A hybrid machine learning approach to network anomaly detection." *Information Sciences*, 177, 2007, 3799–3821.

[2] Kayacik, H. G., Zincir-Heywood, A. N. and Heywood, M. I. "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Benchmark," Proceedings of the PST 2005—International Conference on Privacy, Security, and Trust, pp. 85-89.

[3] Bace, R., Mell, P. "NIST Special Publication on Intrusion Detection Systems." Booz Allen Hamilton Inc. McLean VA, 2001.

[4] TechFAQ. "Responding to Network Attacks and Security Incidents.", http://www.tech-faq.com/responding-to-network-attacks-and-security-incidents.html. Accessed 28 August 2019.

[5] Altwaijry, H., and Algarny, S. "Bayesian Based Intrusion Detection System," *Journal of King Saud University— Computer and Information Sciences*, Vol. 24, No. 1, 2012, pp. 1-6.

[6] Stallings, W. "*Cryptography and network security principles and practices"*. USA: Prentice Hall, 2006.

[7] Olusola., A. A., Oladele, A. S. and Abosede. "Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features." *Proceedings of the World Congress on Engineering and Computer Science I*, San Francisco, 20-22 October 2010.

[8] Sinclair, C., Pierce, L., Matzner, S. "An Application of Machine Learning to Network Intrusion Detection." *In Proceedings of the 15th Annual Computer Security Applications Conference, ACSAC'99*, Washington, DC, USA, 1999.

[9] Sharafaldin, I., Lashkari, A., and Ghorbani, A. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization." *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, January 2018.

[10] Gharib, A., Sharafaldin, I., Lashkari, A., Ghorbani, A. "An Evaluation Framework for Intrusion Detection Dataset." *IEEE*, 2016, pp.1-6.

[11] Calyptix. "Top 8 Network Attacks by Type in 2017", https://www.calyptix.com/top-threats/top-8-network-attacks-type-2017/. Accessed 22 April, 2019

[12] Sadasivam, G. K., Hota, C. and Anand, B. "Classification of SSH Attacks Using Machine Learning Algorithms." *2016 6th International Conference on IT Convergence and Security (ICITCS)*, Prague, pp. 1-6.

[13] Adithya, L., Sampada, K. S. "Segmented File Transfer." *International Journal of Science, Engineering and Computer Technology*, 6(12), 2016, 401.

[14] Khandelwal, S. "Security Risks of FTP and Benefits of Managed File Transfer. The Hacker News.", https://thehackernews.com/2013/12/security-risks-of-ftp-and-benefits-of.html. Accessed 23 August 2019.

[15] Mahoney, M. "Dissertation: A Machine Learning Approach to Detecting Attacks by Identifying Anomalies in Network Traffic.", 2003, pp.1-145.

[16] Zhang and Zulkernine. "A Hybrid Network Intrusion Detection using Random Forests." *Proceedings of the First International Conference on Availability, Reliability and Security*, 2006, pp. 262-269.

[17] Hasan, M., Nasser, M., Pal, B., Ahmad, S. “Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS).” *Journal of Intelligent Learning Systems and Applications,* Vol 6, 2014, pp.45-52.
[18] CICIDS. “Canadian institute for cybersecurity (cic) 2017”, https://www.unb.ca/cic/datasets/ids-2017.html. Accessed 12 February, 2019.
[19] CICFlowMeter (2017). “CICFlowMeter”, http://www.netflowmeter.ca/. Accessed June 25, 2019
[20] Breiman, L. “Random Forests”, *Machine Learning* 45(1), 2001, pp. 5–32.
[21] Guo, Lan, Ma, Yan, Cukic, Bojan and Singh, Harshinder. “Robust Prediction of Fault-Proneness by Random Forests”, *Proceedings of the 15th International Symposium on Software Reliability Engineering (ISSRE'2004)*, pp. 417-428, Brittany, France, November 2004.
[22] Popescu, Bogdan E. and Friedman, Jerome H., “Ensemble Learning for Prediction”, Doctoral Thesis, Stanford University, January 2004.
[23] Wu, Ting-Fan, Lin, Chih-Jen, and Weng, Ruby C. “Probability Estimates for Multi-class Classification by Pairwise Coupling”, *The Journal of Machine Learning Research*, Volume 5, December 2004.
[24] Wu, Yimin. “High-dimensional Pattern Analysis in Multimedia Information Retrieval and Bioinformatics”, Doctoral Thesis, State University of New York, January 2004.
[25] Tharwat, A. (2018). Classification Assessment Methods. Applied Computing and Informatics. Available at: https://doi.org/10.1016/j.aci.2018.08.003
[26] McHugh, J. “Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory.” *ACM Trans. Inf. Syst. Secur*., 3(4), 2000, 262–294.
[27] Brown, C., Cowperthwaite, A., Hijazi, A., and Somayaji, A. “Analysis of the 1999 darpa/lincoln laboratory ids evaluation data.” In *2009 IEEE SCISDA*, pp. 1–7.
[28] Nechaev, B., Allman, M., Paxson, V., and Gurtov, A. (2004). “Lawrence Berkeley national laboratory (lbnl)/icsi enterprise tracing project.”
[29] Li, W., Moore, A. “A Machine Learning Approach for Efficient Traffic Classification.”
Proceedings: MASCOTS '07 Proceedings of the 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 310-317.
[30] Sommer, R., Paxson, V. “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection.” *SP '10 Proceedings of the 2010 IEEE Symposium on Security and Privacy*, pp. 305-316.