Detecting SSH Attacks on Linux Servers Using Machine Learning

Muhammad Usama Zubair
Computer Engineering, Bahria
University, Islamabad, Pakistan

Abstract There are protocols that are used to access cloud servers, Secure Shell (SSH) is the most commonly way to access a secure cloud server. Despite of SSH being 10 years old brute force is still a susceptible to attacks. Our goal is to train and implement a machine learning model which can predict SSH attacks.

Keywords: Secuare Shell,Machine Learning, Cyber-Secuarity, SSH,System Logs

I. Introduction

Secure cloud computing data storage method has changed the way we're technology used to work. Now a day every technology in existence relay on remote servers for computation or data storage in one way or other. Application which were once desktop based now have been move to cloud server, therefore protection of cloud server have become of great importance to use. The most common method used to access these servers by the developer is SSH which is an encrypted way of communication.[1] Even though the SSH encryption is unbreakable there're ways to get access to server through other techniques like SSH brute force attack, dictionary attack and Honey potting, With the increase in computation power brute force method has become more effective than ever before. [2]

With the help of machine learning these patterns of unauthorized accesses can be predicted we're going to use a ML for the purpose of stopping these SSH attacks on the server.

II. Research

When SSH implemented on a server keep a Syslog file documenting every access being made to server. This file can be used to access and view all the connection that are being made to server though SSH. In cyber forensics these files are analyzing to check if a server was hacked or not.[3] However, every time a server is attacked, hacker shred or replace these log file from server making it harder for to analyze and pin point the origin of attack. The most common method which is used by the penetration testers and network security consultants is the overview of log file, an in depth analysis of log files is performed to find the traces of attacks.[2]. KDD99[4] and NSL[5] dataset also provide with the dataset which can be used to predict attacks, but those datasets are were collected in 1999 and aren't a good due to changes in network technology and attack techniques

III. METHODOLOGY

Feature Extraction:

There are not any SSH dataset which can full fil our requirement however several log files are provided on different website by the system administrator logging all these SSH attack.[6] These log files can be found in the system root directory. There are no known system API which let use directly access these log using a programming language however Linux kernel provide us with tail command which let a use see change in the file as soon as the file is update (a new connection is made to sever though SSH) we wrote python based feature extractor which was able to parse following features from the log file.

Feature	Description
ts	Timestamp of when a connection was made to server
is_failure	Is the access being failure or success
ip	IP address of client accessing the server
username	Username being used to access SSH
is_valid	Is Username valid or not
is_root	Is the client trying to sign in as root

Table 1 Primary feature

From these primary features we were able to calculate secondary feature. The secondary features are collected based on the IP address and all the past connection made to the server.

Feature	**Description**
Is_local (Boolean)	Is the IP address of client local or not

Feature	Description
no_failure (Numerical)	Total number of failures since last success login
ip_success (Numerical)	No of Successful login by an IP address
Ip_failure (Numerical)	Total number of failures since last success sign in by an IP address
is_valid (Boolean)	is Username valid or not
is_first (Boolean)	Is it first sign in attempt by a given IP address
not_valid_count (Numerical)	Number of times an invalid username count since last successful login
td (Numerical)	Time difference between two login attempts by an IP
Label	**Description**
class (Boolean)	0 for normal,1 for attack

Table 2 Secondary features

Feature extractor keeps the list of all IP address and in the program memory and calculate these secondary features based on the history by incrementing or decrementing the counters.

Three log files were used to collect the data one of which was create artificially while the other two were generated from real traffic by the servers

Feature Selection:

feature selection determines whether a machine learning model is going to perform good in real word usage or it is going to perform poorly since username and IP address represent static value they can't be used for as machine learning. On the other hand, timestamp was a changing value and had no effect on classification it was also dropped as a feature. Normalizing time difference was the most. EDA of the data set is given below.

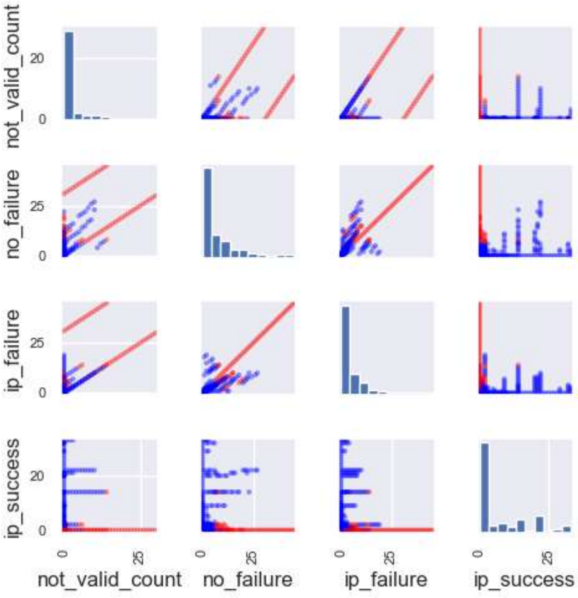

Figure 1 Analysis of features with multiple distinct values

Comprehensive dataset analysis was only performed on select features with multiple (more than two) distinct values. With is_valid, is_failure and is_root was ranked the most significant features (in descending order) during attribute selection.

The features used for this model were as following: *is_private, is_failure, is_root, is_valid, not_valid_count, ip_failure, ip_success, no_failure and first.*

Model Training:

Two classifier and one regressor was used for the model training and testing among which the Gaussian Naïve Bayes be formed the best while Random forest regressor beat Support vector machine by one false negative. Whoever in practical usage both models performed accurately.

Total 267 instance of data was used for the machine learning models. 80% of data was used for training while the rest was used for testing. Confusion matrix for each model was calculated on 20% of training data.

Description	Size
Total Data	267
Train Data	213
Test Data	54

Table 3 Dataset statistics.

Model Improvement:

Each server configured with SSH may have different traffic static or number of users accessing it on per day basis which can lead to False Positive (FP) or False Negative (FN) predictions. we've implemented a technique which improves the model by collecting dataset from the server on which this application is implemented. More over FP and FN can be flag by system administrator which will be used to retrain the model to improve its performance in the network.

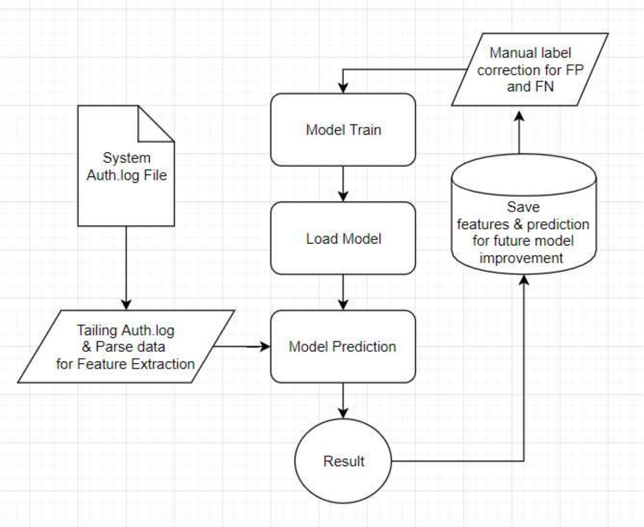
Manual label correction for FP and FN
Model Train
System Auth.log File
Save features & prediction for future model improvement
Load Model
Tailing Auth.log & Parse data for Feature Extraction
Model Prediction
Result

Figure 2 Application architecture

IV. Analysis

When the models are implemented in real word Random Forest Regressor performed slightly better than SVM. Since we were training a linear model it was not possible for SVM to predict instances which were diverting from normal dataset which was being used to train the model. However random forest regressor performed astonishing during real world testing. Therefore, we would recommend using random forest regressor/classifier.

Model	**Predicted Values**			
		X	**0**	**1**
Support Vector Machine	**Actual Values**	**0**	**40**	**0**
		1	**1**	**11**
Random Forest Regressor		**0**	**42**	**0**
		1	**0**	**12**

Table 4 Model behaviour

Test and train accuracies of all the model are given in table 3-CrossValidation was used to find the average value for test and train accuracies.

Model	**Test Accuracy**	**Train Accuracy**
Support Vector Machine (Linear)	99.94%	98.11%
Random Forest Regressor (Estimators = 10)	99.20%	97.13%

Table 5 Train and test accuracies comparison

V. Challenges (Limitations)

There is no dataset available on internet for SSH attack predication except of KDD99 or NSL-KDD (which is improved version of KDD99 dataset).[5] Both datasets have same features which doesn't meet our requirements. Design a feature extractor which can perform parsing from log file in Realtime and keep track of all the previous logs in a data structure was the most difficult part of this process. The initially decision to use time difference as a feature for model training resulted in unacceptable training and test results

Our approach of detecting SSH attack uses IP address to calculate all the secondary features which weigh primary in classifying the incoming SSH request for login if the attacker start using changing IP address before sending every new request model which'll not be able to predict the attack fast enough because it'll have to rely heavily on 'not_valid_count' and 'no_failure' for prediction which can result in inaccurate results.

VI. Conclusion

Linear classification is not fit for SSH attack prediction. Whereas a classification or regression solution based on bagging (an algorithm in which more than one estimator is used to predict the results) technique predicts reliably when implemented on a server. Model improvement can result in decrease in test accuracy.

References

[1] A. Sperotto, R. Sadre, P. T. De Boer, and A. Pras, "Hidden Markov model modeling of SSH brute-force attacks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5841 LNCS, pp. 164–176, 2009.

[2] J. Owens and J. Matthews, "A Study of Passwords and Methods Used in Brute-Force SSH Attacks," *USENIX Work. Large-Scale Exploit. Emergent Threat.*, 2008.

[3] E. Kheirkhah, S. M. P. Amin, H. A. J. Sistani, and H. Acharya, "An experimental study of SSH attacks by using Honeypot Decoys," *Indian J. Sci. Technol.*, vol. 6, no. 12, pp. 5567–5578, 2013.

[4] M. Rummel, "Der Social Entrepreneurship-Diskurs. Eine Einführung in die Thematik," *Wer sind Soc. Entrep. Deutschland?*, no. Cisda, pp. 21–38, 2011.

[5] L. Dhanabal and S. P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms," vol. 4, no. 6, pp. 446–452, 2015.

[6] A. R. Abdou, D. Barrera, and P. C. van Oorschot, "What lies beneath? Analyzing automated SSH bruteforce attacks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9551, pp. 72–91, 2016.