Classification of SSH Attacks using Machine Learning Algorithms

Gokul Kannan Sadasivam, Chittaranjan Hota
Department of Computer Science and Information Systems
BITS, Pilani - Hyderabad Campus
Hyderabad, Telangana, India - 500078
E-mail: {gokul, hota}@hyderabad.bits-pilani.ac.in

Bhojan Anand
School of Computing, National University of Singapore
Computing 1, 13 Computing Drive
Singapore - 117417
E-mail: dcsab@nus.edu.sg

***Abstract*—SSH Attacks are of various types: SSH port scanning, SSH Brute-force attacks, Attacks using compromised SSH server. Attacks using a compromised server could be DoS attacks, Phishing attacks, E-mail spamming and so on. This paper questions whether the attacks from a compromised SSH server be segregated from other attacks using the network flows. In this work, we categorize SSH attacks into two types. The first category consists of all attack activities after a successful compromise of an SSH server. We name it as "severe" attacks. The second type includes all attacks leading to a successful compromise. It consists of SSH port scanning, SSH Brute-force attack, and compromised SSH server with no activities. The second category is named as "not-so-severe" attacks. We employ Machine Learning algorithms, namely, Naive Bayes learner, Logistic Regression, J48 decision tree, and Support Vector Machine to classify these attacks. Suitable features were selected based on domain knowledge, literature survey, and feature selection technique to evaluate the performance of machine learning algorithms using the metrics accuracy, sensitivity, precision, and F-score.**

***Index Terms*—SSH Attacks, Machine Learning, Honeypot, Feature Selection**

I. Introduction

SSH stands for "Secure Shell". It provides remote access to a server, and the interaction is completely protected using standard cryptographic algorithms. The sender encrypts the traffic using a symmetric algorithm like AES. The receiver decrypts the ciphered traffic with the shared key. The shared key can be generated using Diffie-Hellman Key Exchange protocol.

SSH service has been under attack for the past couple of decades. SSH server provides a convenient repository to launch DoS attacks, spread spam messages, and test new malware. A recent survey report [1] has stated that 50% of SSH servers are vulnerable with no updated packages and poor user credentials. In fact, most of the administrators are not technically competent to prevent a misuse from happening.

There are several types of SSH attacks, namely, SSH port scanners, SSH brute-force attacks, SSH penetration attacks, and so on. SSH port scanners are scanning tools used to check if the port 22 is available on any public servers. If the port 22 is open, then the server would respond with a synchronizing message to the attacker. If it is closed, then the server's operating system will send a reset message.

Brute force attacks are used to know the username and password pair of an account on an SSH server. The attacker would try different usernames with different password combinations. Guessing could be done manually or using some automated tools. Automated tools contain a database of commonly used usernames and passwords.

During brute forcing, if an username and password pair is correct, then the attacker gets a login shell with access to the server's filesystem. Consequently, the attacker could do several malicious activities. Some of them are: Injecting a virus program, installing a bot that executes a DDoS attack, spread Spam, or test a newly developed malware. These type of activities are called as penetration attacks.

A typical SSH honeypot (e.g., Kippo SSH Honeypot [2]) segregates network attacks by checking whether the login is successful or not. Kippo manages this by storing all the authenticated connections in separate log files. However, Kippo does not segregate a session based on the network traffic. No methodology had been proposed to separate the flows that are actively used by the attackers. In this paper, we designed a Machine Learning model that classifies not just a successful connection but separates them as severe and not-so-severe.

The remainder of this paper goes as follows: In Section II, related research work concerning the classification of attacks is mentioned. Section III provides four sub-sections detailing the experimental setup, network traffic processing, feature extraction, and class labeling. Section IV details the Machine Learning model with the optimized parameter setting. Section IV also discusses the analyses done for performance evaluation. In Section V, the best set of features is extracted using Exhaustive Search technique. Finally, Section VI concludes the paper with the future scope of work.

II. Related Works

SSH Protocol communication has three sub-protocols: SSH Transport Layer Protocol [3], SSH Authentication Protocol [4], and SSH Connection Protocol [5]. The SSH Transport Layer Protocol is used for negotiating the cryptographic algorithms through unencrypted traffic. The application layer traffic for both SSH Authentication protocol and SSH Connection protocol is encrypted. Hence, packet level inspection in a Firewall is unfeasible.

978-1-5090-3765-0/16/$31.00 ©2016 IEEE

Akihiro Satoh, Yutaka Nakamura, and Takeshi Ikenaga [6] have proposed a methodology to detect the point where SSH Authentication Layer Protocol transitions to SSH Connection Layer Protocol. The two significant features are packet size and packet direction (incoming/outgoing packet). Ward clustering algorithm is used to find this transition point, which further helps to determine if an attack has lead to a successful compromise.

The behavior of SSH Brute-Force attacks can be understood using a Hidden Markov Model [7]. The model consists of seven hidden states: active scanning phase, inactive scanning phase, active brute-force attack phase, inactive brute-force attack phase, active compromise phase, inactive compromise phase, and end state. With the help of flow metrics, the transition probabilities are computed. In "active" states, the attacker is involved in the attack. In "inactive" states, the attacker is temporarily unavailable.

SSHCure [8] [9] is a plugin for NfSen (NetFlow Sensor) tool [10]. It reports SSH attack details, target details, attacker details, and so on. The backend of SSHCure makes use of Flows Per Second (FPS), Packets Per Flow(PPF), and Bytes Per Packet (BPP) to determine SSH attack phases. The purpose of Rick Hofstede et al. [11] work is to improve the existing SSHCure plugin. Certain features in OpenSSH daemon like LoginGraceTime and MaxAuthTries were used to detect SSH compromises. It also takes care of the different attack methodologies executed by hacking tools.

SSH Dictionary attacks are detected using the number of DNS PTR Query packets to a DNS Server [12]. When a dictionary attack occurs on a campus network, there would be a significant amount of DNS PTR Query packets to the local DNS server. The sample variance in the number of DNS Query packet per minute in a time interval of 10 minutes is used to detect dictionary attacks.

Legitimate login failures on SSH server portrays a Beta-Binomial Distribution [13]. When there is a distributed brute-force attack, the Global Factor Indicator (GFI) would deviate from the mean of the normal user traffic distribution. This deviation helps to find distributed brute-force attacks.

Maryam M. Najafabadi et al. [14] used Machine Learning algorithms to detect SSH Brute force attacks. The algorithms used were 5-NN (Nearest Neighbour), C4.5D and C4.5N (Decision Tree algorithms), and Naive Bayes algorithm. Almost all classifiers performed well in classifying the brute force attacks. 5-NN outperformed all other classifiers with and without source port features.

Intrusion Detection Systems also use Machine Learning algorithms to classify attacks. Sandhya Peddabachigari et al. [15] had proposed a Hybrid Intelligent System that classifies attacks into various categories (Denial of Service attacks, Remote to User attacks, User to root attacks, and Probing). Their model uses Decision Tree algorithm, SVM learner, and Hybrid Decision Tree-SVM approach.

III. Data Collection

A. Experimental Setup

A distributed honeypot system was installed and configured in BITS, Pilani - Hyderabad campus [16]. Gokul et al. had built the system on a DELL PowerEdge server. The honeypot comprised low-interaction tools running on different virtual machines. The physical server with a public IP address collected network traffic continuously. The system was active from 22^{nd} July 2014 to 16^{th} August 2014. The honeypot ran incessantly, day and night, without interruption. Kippo SSH Honeypot [2] was used to garner SSH traffic.

The Honeypot system allowed connections on several ports including SSH, MySQL, MS-SQL, and HTTP. Among all the network services, SSH had a significant amount of traffic (approximately 11,000 TCP connections). This traffic comprises of SSH scanning attacks, SSH brute-force attacks, and SSH compromises (successful SSH connections). An in-depth empirical analysis was carried out on SSH traffic with significant statistical inferences.

Their work [16] details the entire setup and its configuration.

B. Network Traffic Processing

The data collected using honeypots comprised of pcap files. These pcap files represent the network traffic between the honeypot and the attackers. The pcap files were processed [17] to get the TCP flow details. A TCP flow is a network communication between two systems (two different IP addresses), sending and receiving packets on a specific port (source port and destination port). The five tuples in a TCP flow are source IP address, destination IP address, source port, destination port, and transport layer protocol.

C. Classes

There are two categories of SSH attacks: severe and not-so-severe. Severe attacks comprise all pcap files which had a successful login attempt followed by execution of a couple of Unix commands. A successful connection could start with a brute force attack. However, it is necessary that the attacker tried, at least, one Unix command in the SSH server.

Not-so-severe attacks comprise the following types.

1) A successful connection with no Unix commands. Aforementioned happens because the attacker is only interested in knowing the login credentials. Once the brute force attack is successful, the connection would be terminated.
2) An unsuccessful connection comprising only login failure messages. It is a typical brute force attack which was unsuccessful.
3) In port scanning attacks, two types were observed.
    a) SYN scanning
    b) Scanning technique that disconnects after receiving the first application layer packet

The authors have manually tagged each of the traffic files with its corresponding labels. Table I provides the list of SSH attacks.

TABLE I
TRAFFIC CLASSES

Severity Level	Traffic Type	Number of TCP flows
Severe Attacks	SSH Compromise with one or more Unix commands	14
Not-So-Severe Attacks	SSH Compromise with no Unix commands	58
	SSH Brute-force attacks	10308
	SSH Port Scanning	552

D. Extraction of features

The TCP flow packets were processed to obtain a set of statistical patterns. The following features were computed from each flow pcap file.

1) Total number of packets
2) Total number of received packets
3) Total number of sent packets
4) The sum of all packet bytes
5) The sum of all received packet bytes
6) The sum of all sent packet bytes
7) The sum of all payload bytes
8) The sum of all received payload bytes
9) The sum of all sent payload bytes
10) Mean inter-arrival time between received packets
11) Variance of inter-arrival time between received packets
12) Total number of packets with ACK flag set
13) Total number of packets with PSH flag set
14) Total number of packets with RST flag set

The range of values in each one of the features are different. Since the algorithm "Logistic Regression" and "Support Vector Machine" performs better with normalised data, each of the features in the data set is normalised using linear scaling and max-min scaling. Features 12, 13, and 14 were normalised using linear scaling. All other features were normalised using max-min normalisation.

IV. MACHINE LEARNING MODEL

The network traffic was captured in pcap file format using the popular "tcpdump" tool. The pcap file was completely processed to remove unwanted network traffic. This traffic includes Ubuntu updates/upgrades, legitimate traffic, and honeypot status messages. Splitcap tool [17] was used to separate the merged pcap file into individual TCP flows.

A Java program interfacing with jnetpcap library [18] was written to extract relevant features from each flow pcap file. Some of the characteristics (mean inter-arrival time, the variance of inter-arrival time) were computed using statistical formulae. Weka tool [19] was used run the Machine Learning algorithms.

A. Machine Learning Algorithms

Four Machine Learning Supervised algorithms are chosen to evaluate the performance of classification. The four algorithms are Naive Bayes (NB) algorithm, Logistic Regression (LR), J48 Decision Tree (J48), and Support Vector Machine (SVM). Logistic Regression and SVM performs well for a dataset with only numerical attributes. Decision Tree is one of the widely used algorithms in the field of Information Security. Even though Naive Bayes algorithm is well suited for nominal attributes, it can be applied even for numerical attributes. Also, ZeroR is used as a baseline classifier.

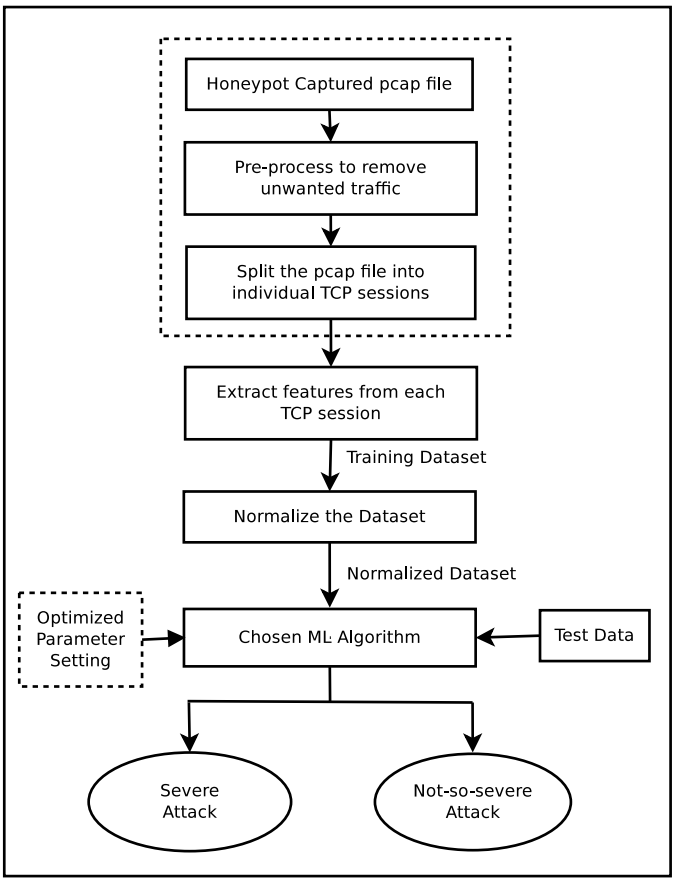

Fig. 1. Machine Learning Model

Naive Bayes algorithm [20] utilizes the conditional probability formula derived from Bayes Theorem. Naive Bayes helps in predicting the probability of occurrence of all classes and chooses the one with the highest probability. When the features are numerical, each feature can be treated as a Gaussian distribution. The features in this work portray a Gaussian distribution.

Logistic Regression uses a sigmoid function to determine the right output class. In Equation 1, L is the cost function; Y_i is the binary output value; P_{SA} and P_{NSA} stand for the probability of a severe and that of a not-so-severe attack, respectively; X_i is a normalised data sample; r is the ridge parameter; B is the Logistic Regression coefficients. The ridge parameter is 1×10^{-8}. A Quasi-Newton method finds the optimal values of B.

TABLE II
CONFUSION MATRIX

	Predicted: Severe Attack	**Predicted: Not-So-Severe Attack**
Actual: Severe Attack	TS	FNS
Actual: Not-So-Severe Attack	FS	TNS

$$L = -\sum_{i=1}^{n} (Y_i \log P_{SA}(X_i) + (1 - Y_i) \log P_{NSA}(X_i)) + rB^2 \quad (1)$$

J48 is a Java version of the popular decision tree algorithm C4.5 (Revision 8). The minimum number of instances per leaf is 10. The pruned and the unpruned version produced the same results because the amount of misclassification was very low. For the pruned version, the confidence factor was 0.25.

SVM is a binary linear classifier that outputs an hyperplane with a large margin between the positive and the negative samples. Sequential Minimal Optimization (SMO) [21] computes the coefficients of the SVM algorithm. The value of epsilon that reduces bias is 1 x 10^{-12}. The kernel method is a normalized polynomial function with the degree of the polynomial as 24. In Equation 2, K' is a kernel function, x and y are the data samples, and d is the degree of the polynomial. In Equation 3, K is the normalized kernel function.

$$K'(x, y) = [x.y]^d = [x.y]^{24} \quad (2)$$

$$K(x, y) = \frac{K'(x, y)}{\sqrt{K'(x, x).K'(y, y)}} \quad (3)$$

In all the above algorithms, 10-fold stratified cross-validation was employed to get the output. In each fold, the dataset was randomly chosen. Before application of each fold, the dataset was randomized again.

B. Performance of ML Algorithms

In this section, we describe the performance of the chosen set of algorithms. For each algorithm, the Confusion Matrix is computed. The Confusion Matrix for our problem is modified, as shown in Table II.

In Table II, TS, FS, TNS, and FNS represents True Severe Attack, False Severe Attack, True Not-So-Severe Attack, and False Not-So-Severe Attack, respectively. The confusion matrix for the five ML algorithms is shown in Table III.

With the help of the Confusion Matrix, the following performance metrics are computed.

TABLE III
CONFUSION MATRIX FOR ML ALGORITHMS

Algorithm	**TS**	**FS**	**TNS**	**FNS**
ZeroR	0	0	10918	14
NB	12	5	10913	2
LR	10	4	10914	4
J48	13	0	10918	1
SVM	8	3	10915	6

TABLE IV
PERFORMANCE EVALUATION

Algo.	**acc.**	**sens.**	**prec.**	**F_2 score**
ZeroR	0.9987	0.0000	-	-
NB	0.9994	0.8571	0.7059	0.8219
LR	0.9993	0.7143	0.7143	0.7143
J48	0.9999	0.9286	1.0000	0.9420
SVM	0.9992	0.5714	0.7273	0.5970

$$\text{Accuracy (acc.)} = \frac{TS + TNS}{TS + TNS + FS + FNS} \quad (4)$$

$$\text{Sensitivity (sens.)} = \frac{TS}{TS + FNS} \quad (5)$$

$$\text{Precision (prec.)} = \frac{TS}{TS + FS} \quad (6)$$

$$\text{F-score} = \frac{(1 + \beta^2)(prec. \times sens.)}{(\beta^2 prec.) + sens.} \quad (7)$$

The accuracy (as given in Equation 4) provides the ratio between the attacks correctly classified to the overall attacks. Here, attacks refer to both severe and not-so-severe. A Machine Learning algorithm that provides a low value of accuracy will also have a low value for sensitivity and precision. Hence, accuracy is the first performance metric that should be satisfied by a Machine Learning algorithm. As shown in Table IV, the accuracy is more than 99% for all the chosen algorithms. Neglecting ZeroR all other algorithms are eligible candidates for performance comparison.

Sensitivity (as provided by Equation 5) is the fraction of severe attacks that are correctly predicted by a classifier. Compared to the set of algorithms, J48 decision tree had the highest sensitivity of 92.86%. Naive Bayes Algorithm and Logistic Regression were comparatively better than Support Vector Machine, which was able to detect only 57.14% of all the severe attacks.

Precision (as provided by Equation 6) conveys the fraction of predictions that are correct. J48 predictions are very accurate. As seen in column 4 of Table IV, the precision of the J48 algorithm hit the maximum of 100%. The performance of SVM is minutely better than Naive Bayes and Logistic Regression. Naive Bayes, which had a better sensitivity, has a relatively poor precision.

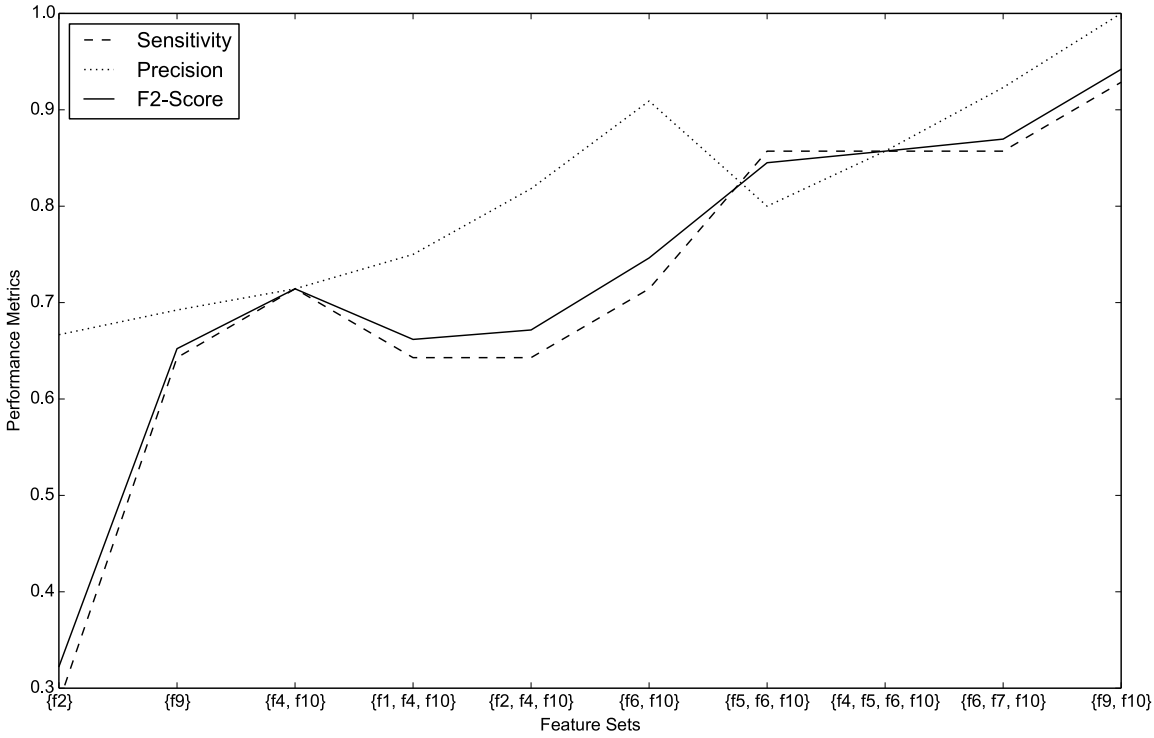

Fig. 2. NB Best Feature Set Performance

F-score (as given in Equation 7) is the weighted average of sensitivity and precision. The value of β determines the balance between sensitivity and precision. When $\beta > 1$, the F-score gives more preference to sensitivity than precision. In this work, sensitivity is given more importance than precision. Hence, the authors have chosen a β of 2. F-score reaches its best value at 1 and worst value at 0. As previously observed, the J48 decision tree algorithm had the maximum F_2 score. Irrespective of the poor precision value by Naive Bayes algorithm, it had a better F_2 score. It is due to the balance provided by its sensitivity value. The F_2 score of Logistic Regression remained the same as its sensitivity and precision. SVM had an approximate value close to its sensitivity.

TABLE V
BEST FEATURE SET USING NAIVE BAYES

Best Feature Set	Accuracy
2	0.99894
9	0.99914
4 10	0.99923
1 4 10	0.99934
2 4 10	0.99943
6 10	0.99954
5 6 10	0.99963
4 5 6 10	0.99969
6 7 10	0.99973
9 10	0.99991

V. FEATURE SELECTION

Feature Selection [22] is a process of determining the redundant and unnecessary features in a data set. It reduces the computational cost involved in training a data set. It also eliminates data overfitting. By decreasing the bias, it helps achieve more accuracy. The three standard techniques are Subset selection (Wrapper method), Filter method and Embedded approach. All techniques use heuristics to estimate the best features.

Exhaustive search is a Wrapper method that tries each combination of features to obtain the best feature set. Hence, it is guaranteed to produce the correct answer. However, the downside is the computation time required. Given the dimensionality of our data set, it is feasible to use this technique to obtain the best features.

Naive Bayes is the performance evaluation algorithm utilized in the exhaustive search. Exhaustive search technique produced the best feature set in the order mentioned in Table V, with accuracy as the performance metric.

Accuracy alone is not sufficient to decide on the best feature set. Hence, sensitivity, precision, and F-score were computed to find the best feature set. Figure 2 shows the performance metrics for various feature sets generated by the feature selection technique. From Figure 2, it could be inferred

TABLE VI
PERFORMANCE OF LR AND SVM

	LR	**SVM**
Accuracy	0.9997	0.9990
Sensitivity	0.9286	1.0000
Precision	0.8667	0.5600
F_2-Score	0.9155	0.8642

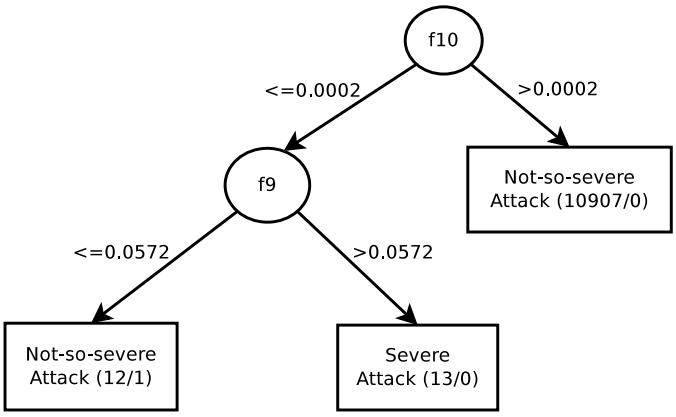

Fig. 3. J48 Decision Tree

that the set comprising features 9 and 10 produces an optimal performance.

The features f9 and f10 are used to evaluate the performance of Logistic Regression algorithm and SVM algorithm. The accuracy, sensitivity, precision, and F_2 score have improved prominently for Logistic Regression algorithm (as shown in Table VI). On the other hand, SVM had a slight dip in accuracy and precision. The decrease was due to the misclassification of 14 different not-so-severe attacks. Importantly, SVM classifies all the severe attacks leading to a sensitivity of 100%.

Figure 3 shows the pruned decision tree generated by the J48 algorithm. The feature f10 corresponds to the mean inter-arrival time between received packets. The feature f9 corresponds to the sum of all sent payload bytes. As shown in Figure 3, feature f10 was able to segregate most of the not-so-severe attacks (10907 attacks). Feature f9 did justice in separating most of the severe attacks (13 attacks). If the mean inter-arrival time of received packets is greater than 1651 milliseconds, then it is highly likely that the attack is not-so-severe. If there are more than 3216 sent payload bytes in a session, then it would probably be a severe attack.

VI. CONCLUSION

We have proposed a model to classify SSH attacks based on severity level. We found that feature set {f9, f10} are sufficient to classify severity of SSH attacks. We did in-depth analysis on the performance of several Machine Learning algorithms. The performance of J48 seemed to be promising and produced better results. In future, we will analyse the performance of ML algorithms in classifying attacks in real-time.

REFERENCES

[1] (2014) Ssh security vulnerability report. [Online]. Available: https://www.venafi.com/assets/pdf/Ponemon_2014_SSH_Security_Vulnerability_Report.pdf

[2] Kippo ssh honeypot. [Online]. Available: https://github.com/desaster/kippo

[3] (2016) Ssh transport layer protocol. [Online]. Available: https://www.ietf.org/rfc/rfc4253.txt

[4] (2016) Ssh authentication protocol. [Online]. Available: https://www.ietf.org/rfc/rfc4252.txt

[5] (2016) Ssh connection protocol. [Online]. Available: https://www.ietf.org/rfc/rfc4254.txt

[6] A. Satoh, Y. Nakamura, and T. Ikenaga, "Ssh dictionary attack detection based on flow analysis," in *Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on*, July 2012, pp. 51–59.

[7] A. Sperotto, R. Sadre, P.-T. Boer, and A. Pras, "Hidden markov model modeling of ssh brute-force attacks," in *Proceedings of the 20th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management: Integrated Management of Systems, Services, Processes and People in IT*, ser. DSOM '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 164–176. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04989-7_13

[8] (2016) Sshcure. [Online]. Available: http://sshcure.sf.net

[9] L. Hellemons, L. Hendriks, R. Hofstede, A. Sperotto, R. Sadre, and A. Pras, "Sshcure: a flow-based ssh intrusion detection system," in *Dependable Networks and Services*, ser. Lecture Notes in Computer Science, vol. 7279. Berlin, Germany: Springer Verlag, June 2012, pp. 86–97. [Online]. Available: http://doc.utwente.nl/80710/

[10] (2011) Nfsen. [Online]. Available: http://nfsen.sourceforge.net/

[11] R. Hofstede, L. Hendriks, A. Sperotto, and A. Pras, "Ssh compromise detection using netflow/ipfix," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 20–26, Oct. 2014. [Online]. Available: http://doi.acm.org/10.1145/2677046.2677050

[12] M. Kumagai, Y. Musashi, D. A. L. Romana, K. Takemori, S. Kubota, and K. Sugitani, "Ssh dictionary attack and dns reverse resolution traffic in campus network," in *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*, Nov 2010, pp. 645–648.

[13] M. Javed and V. Paxson, "Detecting stealthy, distributed ssh brute-forcing," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 85–96. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516719

[14] M. Najafabadi, T. Khoshgoftaar, C. Kemp, N. Seliya, and R. Zuech, "Machine learning for detecting brute force attacks at the network level," in *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*, Nov 2014, pp. 379–385.

[15] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114 – 132, 2007, network and Information Security: A Computational Intelligence ApproachNetwork and Information Security: A Computational Intelligence Approach. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804505000445

[16] G. Sadasivam and C. Hota, "Scalable honeypot architecture for identifying malicious network activities," in *Emerging Information Technology and Engineering Solutions (EITES), 2015 International Conference on*, Feb 2015, pp. 27–31.

[17] Netresec. (2013) Splitcap tool. [Online]. Available: http://www.netresec.com/?page=SplitCap

[18] jnetpcap library. [Online]. Available: http://jnetpcap.com

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[20] *Machine Learning*. McGraw-Hill, 1997.

[21] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Research, Tech. Rep. MSR-TR-98-14, April 1998. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=69644

[22] *Introduction to Data Mining*. PEARSON, 2006.