Classification of SSH Anomalous Connections

Silvia González[1], Javier Sedano[1], Urko Zurutuza [2], Enaitz Ezpeleta[2], Diego Martínez[3], Álvaro Herrero[3], and Emilio Corchado[4]

[1]Instituto Tecnológico de Castilla y León
C/ López Bravo 70, Pol. Ind. Villalonquejar, 09001 Burgos, Spain
javier.sedano@itcl.es

[2]Electronics and Computing Department, Mondragon University
Goiru Kalea, 2, 20500 Arrasate-Mondragon, Spain
{uzurutuza,eezpeleta}@mondragon.edu

[3]Department of Civil Engineering, University of Burgos, Spain
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
ahcosio@ubu.es

[4]Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced, s/n, 37008 Salamanca, Spain
escorchado@usal.es

Abstract. The Secure Shell Protocol (SSH) is a well-known standard protocol for remote login and used as well for other secure network services over an insecure network. It is mainly used for remotely accessing shell accounts on Unix-liked operating systems to perform administrative tasks. For this reason, the SSH service has been for years an attractive target for attackers, aiming to guess root passwords performing dictionary attacks, or to directly exploit the service itself. To test the classification performance of different classifiers and combinations of them, this study gathers and analyze SSH data coming from a honeynet and then it is analysed by means of a wide range of classifiers. The high-rate classification results lead to positive conclusions about the identification of malicious SSH connections.

Keywords: Secure Shell Protocol, SSH, Honeynet, Honeypot, Intrusion Detection, Classifier, Ensemble

1 Introduction

A network attack or intrusion will inevitably violate one of the three computer security principles -availability, integrity and confidentiality- by exploiting certain vulnerabilities such as Denial of Service, Modification and Destruction [1]. One of the most harmful issues of attacks and intrusions, which increases the difficulty of protecting computer systems, is precisely the ever-changing nature of attack technologies and strategies.

Intrusion Detection Systems (IDSs) [2-4] have become an essential asset in addition to the computer security infrastructure of most organizations. In the context of computer networks, an IDS can roughly be defined as a tool designed to detect suspicious patterns that may be related to a network or system attack. Intrusion Detection (ID) is therefore a field that focuses on the identification of attempted or ongoing attacks on a computer system (Host IDS - HIDS) or network (Network IDS - NIDS).

ID has been approached from several different points of view up to now; many different Computational Intelligence techniques - such as Genetic Programming [5], Data Mining [6-8], Expert Systems [9], Fuzzy Logic [10], or Neural Networks [11-13] among others - together with statistical [14] and signature verification [15] techniques have been applied mainly to perform a 2-class classification (normal/anomalous or intrusive/non-intrusive).

The Secure Shell Protocol (SSH) is a standard protocol for remote login and used as well for other secure network services over an insecure network. It is an Application Layer protocol under the TCP/IP stack. The SSH protocol consists of three major components: The Transport Layer Protocol that provides server authentication, confidentiality, and integrity with perfect forward secrecy. The User Authentication Protocol which authenticates the client to the server. And the Connection Protocol that multiplexes the encrypted tunnel into several logical channels.

The main usage of SSH protocol is for remotely accessing shell accounts on Unix-liked operating systems with administrative purposes. For this reason, the SSH service has been for years an attractive service for attackers, aiming to guess root passwords performing dictionary attacks, or to directly exploit the service itself. The SANS Institute's Internet Storm Center [16] keeps monitoring an average of 100,000 targets being attacked every day in Internet. Being able of distinguishing among malicious SSH packets and benign SSH traffic for server administration may play an indispensable role in defending system administrators against malicious adversaries.

The aim of the present study is to assess classifiers and ensembles in the useful task of identifying bad-intentioned SSH connections. To do so, real data, coming from the Euskalert honeynet is analysed as described in the remaining sections of the paper. In this contribution, section 2 presents the proposed models that are applied to SSH data as described in section 3, together wit the obtained results. Some conclusions and lines of future work are introduced in section 4.

1.1 SSH and Honeynets

A honeypot has no authorised function or productive value within the corporate network other than to be explored, attacked or compromised [17]. Thus, a honeypot should not receive any traffic at all. Any connection attempt with a honeypot is then an attack or attempt to compromise the device or services that it is offering- is by default illegitimate traffic. From the security point of view, there is a great deal of information that may be learnt from a honeypot about a hacker's tools and methods in order to improve the protection of information systems.

In a honeynet, all the traffic received by the sensors is suspicious by default. Thus every packet should be considered as an attack or at least as a piece of a multi-step

attack. Numerous studies propose the use of honeypots to detect automatic large scale attacks; honeyd [18] and nepenthes [19] among others. The first Internet traffic monitors known as Network Telescopes, Black Holes or Internet Sinks were presented by Moore *et al*. [20].

The Euskalert honeynet [21] has been monitoring attacks against well-known services, including SSH. Furthermore, the sensors have recorded the SSH sessions used to administer and maintain the different devices of the infrastructure.

Having both malicious and real administrative SSH traffic recorded, we perform a classification of such traffic to detect attacks against the SSH service.

1.2 Previous Work

Attacks to SSH service have attracted researchers' attention for a long time. Song et al. [22] analysed timing and keystroke attacks. Researchers have also used honeypots to study and analyse attacks to this protocol, focusing on login attempts and dictionary attacks [23], [24]. In [24] authors analyse SSH attacks on honeypots focusing on visualisation of the data gathered. The honeypots collect real attacks, making experiments and analysis results applicable to real deployments.

Considering the data capture, as previously introduced, the present study takes advantage of the Euskalert project [21]. It has deployed a network of honeypots in the Basque Country (northern Spain) where eight companies and institutions have installed one of the project's sensors behind the firewalls of their corporate networks. The honeypot sensor transmits all the traffic received to a database via a secure communication channel. These partners can consult information relative to their sensor (after a login process) as well as general statistics in the project's website. Once the system is fully established, the information available can be used to analyse attacks suffered by the honeynet at network and application level. Euskalert is a distributed honeypot network based on a Honeynet GenIII architecture [25].

2 Proposal

One of the most interesting features of IDSs would be their capability to automatically detect whether a portion of the traffic circulating the network is an attack or normal traffic. This task is more challenging when confronting brand-new bad intentioned activities with no previous examples. Automated learning models (classifiers) [26] are well-known algorithms designed specifically for the purpose of deciding about previously-unseen data. This issue makes them suitable for the IDS task. Going one step further, ensemble methods [27] combine multiple algorithms into one usually more accurate than the best of its components. So, the main idea behind ensemble learning is taking advantage of classification algorithms diversity to face more complex data. For this reason, present study proposes the combination of classifiers to get more accurate results when detecting anomalous and intrusive events.

A wide variety of automated learning techniques have been applied in this study to classify SSH connections. Several base classifiers as well as different ways of combining them have been considered for the analysis of Euskalert data. 35 base classifi-

ers have been applied in present study, comprising neural models such as the Multi-Layer Perceptron and Voted-Perceptron [28], decision trees such as CART [31] or REP-Tree [32], and traditional clustering algorithms such as the k-Nearest Neighbours (K-NN) [30].

These base classifiers have been combined according to the ensemble paradigm by 19 different strategies. The applied ensemble schemes range from basic ones such as Bagging [33] or boosting-based [34] (Adaboost) to some other, more modern algorithms such as the LogitBoost [35] or the StackingC [36]. As results prove, ensemble learning adds an important value to the analysis, as almost all variants consistently improve results obtained by the single classifier.

3 Experimental Validation on Real Data

As previously mentioned, the performance of automated learning techniques have been assessed using real datasets, coming from the Euskalert project. The detailed information about the data and the run experiments is provided in this section.

3.1 Datasets

We have performed the experimental study by extracting SSH data related to 34 months of real attacks and administration tasks that reached the 8 sensors of the Euskalert project [25]. Data from a so long time period guarantees that a broad variety of situations are considered.

This honeynet system receives 4,000 packets a month on average. The complete dataset contains a total of 2,647,074 packets, including TCP, UDP and ICMP traffic received by the distributed honeypot sensors. For this experiment, we have analysed SSH connections happened between May 2008 and March 2011. First, we have filtered out traffic containing real attacks to the SSH port (22), and SSH connections to the system management port (2399).

Then, the traffic has been processed in order to obtain the Secure Shell sessions out of the packets. Two different approaches have been used in order to identify the sessions:

The approach for defining an SSH session was based on the TCP logic, using packets with the same source IP, same destination IP and a common source port. This last value is a non-privileged port number that remains the same during any TCP session. Out of the 2,647,074 packets, the TCP-based dataset was summarized as 8,478 attack sessions and 82 administration sections.

The features that were extracted from each one of the sessions in the dataset are described in table 1.

Table 1. Features for SSH sessions.

Feature	Description
Src	IP address of the source host

Time	duration of the session
Numpac	number of packets that the source host sent
Minlen	minimum size of the packets
Maxlen	maximum size of the packets
Avglen	average size of the packets
Numflags	amount of different flags used

Table 2 shows the range of each feature, depending on the nature of the session (administrator or attack).

Table 2. Range of features for SSH sessions.

Feature	Type	Attack	Administrator
Src	inet	---	---
Time	interval	00:00:00 – 352 days 09:48:19.891	00:00:00.004 – 519 days 18:24:05.446
Numpac	integer	1 - 95	1 - 23
Minlen	integer	40 - 64	40 - 380
Maxlen	integer	40 - 220	40 - 380
Avglen	numeric(8,2)	40 - 96	40 - 380
Numflags	integer	1 - 6	1 - 4

3.2 Practical Settings

The experimentation has been based on the performance of 1,534 tests, carried out through 35 different classifiers (such as "NaiveBayes", "Ibk", "LinearRegression", "JRip", "RBFNetwork", "SMO", etc.) combined by means of the following ensembles: Base classifier, Bagging, Adaboost, MultiBoostAB, RandomSubSpace, Dagging, Decorate, MultiClassClassifier, CVParameterSelection, AttributeSelectedClassifier, ThresholdSelector, Vote, FilteredClassifier, Grading, MultiScheme, OrdinalClassClassifier, RotationForest, Stacking, and StackingC.

For testing purposes, each ensemble processes a combination of 10 same type base classifiers. The data sets were trained and classified with ensembles and classifiers by means of WEKA software [37].

3.3 Results

To summarize the results data, only the classification rate from the base classifier and the highest rate from the different ensembles are shown below in Table 3 (comprising training results) and Table 4 (comprising classification results).

Table 3. Training results on SSH sessions.

#	Classifier	Base Classifier	Max
1	MultilayerPerceptron	0,99325	0,998053
2	Naive Bayes	0,992861	0,99325
3	K-nn IBK	0,997793	0,997923
4	Decision tree SImpleCart	0,993899	0,998053
5	Rule Induction Jrip	0,995846	0,998183
6	RBF network	0,994159	0,996495
7	REPTree	0,995717	0,997274
8	NaiveBayesMultinomial	0,870457	0,990395
9	IB1	0,997923	0,997923
10	PART	0,996885	0,997923
11	ZeroR	0,990395	0,990395
12	BayesianLogisticRegression	0,990395	0,991044
13	ComplementNaiveBayes	0,754154	0,990395
14	DMNBtext	0,990395	0,990395
15	NaiveBayesMultinomialUpdateable	0,872144	0,990395
16	NaiveBayesUpdateable	0,992861	0,99325
17	Logistic	0,992082	0,997534
18	SMO	0,990524	0,998183
19	SPegasos	0,990395	0,990395
20	VotedPerceptron	0,990395	0,997664
21	DTNB	0,997534	0,998053
22	DecisionTable	0,998053	0,998183
23	NNge	0,995976	0,997274
24	OneR	0,997534	0,997793
25	Ridor	0,996625	0,997534
26	ADTree	0,996366	0,998183
27	BFTree	0,99338	0,998183
28	DecisionStump	0,99351	0,995457
29	FT	0,994548	0,997793
30	J48	0,995846	0,998183
31	LADTree	0,995846	0,997923
32	LMT	0,995067	0,997923
33	NBTree	0,994029	0,997923
34	RandomForest	0,996885	0,997793
35	RandomTree	0,996495	0,997404

Table 4. Classification results on SSH sessions.

#	Classifier	Base Classifier	Max
1	MultilayerPerceptron	0,992982	0,997661
2	Naive Bayes	0,992982	0,992982
3	K-nn IBK	0,996491	0,997661
4	Decision tree SImpleCart	0,994152	0,997661
5	Rule Induction Jrip	0,995322	0,997661
6	RBF network	0,992982	0,996491
7	REPTree	0,994152	0,997661
8	NaiveBayesMultinomial	0,854971	0,990643
9	IB1	0,996491	0,997661
10	PART	0,997661	0,99883
11	ZeroR	0,990643	0,990643
12	BayesianLogisticRegression	0,990643	0,991813
13	ComplementNaiveBayes	0,753216	0,990643
14	DMNBtext	0,990643	0,990643
15	NaiveBayesMultinomialUpdateable	0,85614	0,990643
16	NaiveBayesUpdateable	0,992982	0,992982
17	Logistic	0,991813	0,997661
18	SMO	0,990643	0,997661
19	SPegasos	0,990643	0,997661
20	VotedPerceptron	0,990643	0,997661
21	DTNB	0,997661	0,997661
22	DecisionTable	0,997661	**1**
23	NNge	0,996491	0,997661
24	OneR	0,997661	0,99883
25	Ridor	0,997661	0,997661
26	ADTree	0,995322	0,997661
27	BFTree	0,994152	0,997661
28	DecisionStump	0,992982	0,995322
29	FT	0,992982	0,997661
30	J48	0,995322	0,99883
31	LADTree	0,994152	0,997661
32	LMT	0,991813	0,997661
33	NBTree	0,992982	0,997661
34	RandomForest	0,997661	0,997661
35	RandomTree	0,997661	0,99883

From Table 4, it can be seen that the best classification result (1) is obtained by the DecisionTable classifier combined by the Adaboost ensemble.

4 Conclusions and Future Work

Classification of benign and malicious SSH sessions is extremely valuable for preventing unauthorized users to access production networks. The successful classification results obtained in this study can efficiently discover a malicious connection attempt and make possible to discard the session before a dictionary attack becomes a major problem to the network assets.

It has been shown how base classifiers provide good results in differentiating real administering SSH sessions from attacks, but the use of ensemble classifiers even improve the effectiveness up to a 100% in at least one case.

This may be due to the fact that a real SSH session can be comprised by an increasing number of different bash commands, generated by few different IP addresses (administrators). This would derive in a more specific behaviour than the rest of SSH attacks gathered by the honeynet.

Those exceptional classification results obtained by ensemble classifiers can be applied to other protocols and services of the attacks received by the honeynets, such as HTTP, SNMTP, or even FTP, learning from the honeypots classification models that will later prevent detected attacks surpass the organization networks causing any damage.

Acknowledgments. This research is partially supported through projects of the Spanish Ministry of Economy and Competitiveness with ref: TIN2010-21272-C02-01 (funded by the European Regional Development Fund), and SA405A12-2 from Junta de Castilla y León.

References

1. Myerson, J.M.: Identifying Enterprise Network Vulnerabilities. International Journal of Network Management 12 (2002) 135-144
2. Computer Security Threat Monitoring and Surveillance. Technical Report. James P. Anderson Co (1980)
3. Denning, D.E.: An Intrusion-Detection Model. IEEE Transactions on Software Engineering 13 (1987) 222-232
4. Chih-Fong, T., Yu-Feng, H., Chia-Ying, L., Wei-Yang, L.: Intrusion Detection by Machine Learning: A Review. Expert Systems with Applications 36 (2009) 11994-12000
5. Abraham, A., Grosan, C., Martin-Vide, C.: Evolutionary Design of Intrusion Detection Programs. International Journal of Network Security 4 (2007) 328-339
6. Julisch, K.: Data Mining for Intrusion Detection: A Critical Review. In: Barbará, D., Jajodia, S. (eds.): Applications of Data Mining in Computer Security. Kluwer Academic Publishers (2002) 33-62
7. Giacinto, G., Roli, F., Didaci, L.: Fusion of Multiple Classifiers for Intrusion Detection in Computer Networks. Pattern Recognition Letters 24 (2003) 1795-1803
8. Chebrolu, S., Abraham, A., Thomas, J.P.: Feature Deduction and Ensemble Design of Intrusion Detection Systems. Computers & Security 24 (2005) 295-307

9. Kim, H.K., Im, K.H., Park, S.C.: DSS for Computer Security Incident Response Applying CBR and Collaborative Response. Expert Systems with Applications 37 (2010) 852-870
10. Tajbakhsh, A., Rahmati, M., Mirzaei, A.: Intrusion Detection using Fuzzy Association Rules. Applied Soft Computing 9 (2009) 462-469
11. Sarasamma, S.T., Zhu, Q.M.A., Huff, J.: Hierarchical Kohonenen Net for Anomaly Detection in Network Security. IEEE Transactions on Systems Man and Cybernetics, Part B 35 (2005) 302-312
12. Herrero, Á., Corchado, E., Gastaldo, P., Zunino, R.: Neural Projection Techniques for the Visual Inspection of Network Traffic. Neurocomputing 72 (2009) 3649-3658
13. Zhang, C., Jiang, J., Kamel, M.: Intrusion Detection using Hierarchical Neural Networks. Pattern Recognition Letters 26 (2005) 779-791
14. Marchette, D.J.: Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint. Springer-Verlag New York, Inc. (2001)
15. Roesch, M.: Snort–Lightweight Intrusion Detection for Networks. 13th Systems Administration Conference (LISA '99) (1999) 229-238
16. SANS Institute's Internet Storm Center https://isc.sans.edu/port.html?port=22
17. Charles, K.A.: Decoy Systems: A New Player in Network Security and Computer Incident Response. International Journal of Digital Evidence 2 (2004)
18. Provos, N.: A Virtual Honeypot Framework. 13th USENIX Security Symposium, Vol. 132 (2004)
19. Baecher, P., Koetter, M., Holz, T., Dornseif, M., Freiling, F.: The Nepenthes Platform: An Efficient Approach to Collect Malware. 9th International Symposium on Recent Advances in Intrusion Detection (RAID 2006), Vol. 4219. Springer Berlin / Heidelberg (2006) 165-184
20. Moore, D., Shannon, C., Brown, D.J., Voelker, G.M., Savage, S.: Inferring Internet Denial-of-service Activity. ACM Transactions on Computer Systems 24 (2006) 115-139
21. Herrero, Á., Zurutuza, U., Corchado, E.: A Neural-Visualization IDS for Honeynet Data. International Journal of Neural Systems 22 (2012) 1-18
22. Song, D.X., Wagner, D., Tian, X.: Timing Analysis of Keystrokes and Timing Attacks on SSH. Proceedings of the 10th conference on USENIX Security Symposium, Vol. 10. USENIX Association, Washington, D.C. (2001) 25-25
23. Coster, D.D., Woutersen, D.: Beyond the SSH Brute Force Attacks. 10th GOVCERT.NL Symposium (2011)
24. Koniaris, I., Papadimitriou, G., Nicopolitidis, P.: Analysis and Visualization of SSH Attacks Using Honeypots. IEEE European Conference on Computer as a Tool (IEEE EUROCON 2013) (2013)
25. Friedman, J.H., Tukey, J.W.: A Projection Pursuit Algorithm for Exploratory Data-Analysis. IEEE Transactions on Computers 23 (1974) 881-890
26. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2007)
27. Seni, G., Elder, J.: Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. Morgan and Claypool Publishers (2010)
28. Freund, Y., Schapire, R.E.: Large Margin Classification Using the Perceptron Algorithm. Mach. Learn. 37 (1999) 277-296
29. Moody, J., Darken, C.J.: Fast Learning in Networks of Locally-tuned Processing Units. Neural computation 1 (1989) 281-294
30. Bailey, T., Jain, A.: A Note on Distance-Weighted k-Nearest Neighbor Rules. IEEE Transactions on Systems, Man and Cybernetics 8 (1978) 311-313
31. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth Inc., Belmont, CA 358 (1984)
32. Zhao, Y., Zhang, Y.: Comparison of Decision Tree Methods for Finding Active Objects. Advances in Space Research 41 (2008) 1955-1959
33. Breiman, L.: Bagging Predictors. Machine Learning 24 (1996) 123-140

34. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. International Conference on Machine Learning (1996) 148-156
35. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. The Annals of Statistics 28 (2000) 337-407
36. Seewald, A.K.: How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness. Nineteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. (2002)
37. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. ACM SIGKDD Explorations Newsletter 11 (2009) 10-18