Poster: Method for Effective Measurement, Labeling, and Classification of Botnet C2s for Predicting Attacks

Mitsuhiro Hatada
National Institute of Standards and Technology
mitsuhiro.hatada@nist.gov

Matthew Scholl
National Institute of Standards and Technology
matthew.scholl@nist.gov

***Abstract*—In the era of the Internet of Things, botnet threats are rising, which has prompted many studies on botnet detection. This study aims to detect the early signs of botnet attacks such as massive spam emails and Distributed Denial-of-Service attacks. To that end, this study develops a practical method for measurement, labeling, and classification of botnet Command and Control (C2) for predicting attacks. The focus is on C2 traffic and measurement of the comprehensive metrics studied in previous works. The data is labeled based on the result of the correlation analysis between C2 metrics and spam volume. Then, a special type of recurrent neural network, i.e., Long Short-Term Memory, is applied to detect an increase in spam by a botnet. The proposed method managed to detect it with an accuracy of 0.981.**

I. Introduction

A botnet is still a serious threat to cybersecurity as it controls a massive number of compromised hosts to conduct various attacks such as sending email spam or launching a Distributed Denial-of-Service (DDoS) attack. A Command and Control (C2) server plays a significant role in a botnet: it sends commands to bots and receives outputs of bots while hiding a botmaster behind it. Despite many previous attempts at botnet measurement [6], [8] and botnet detection [4], [5], [7], [10], to the best of our knowledge, there is no study that *detects the early signs of botnet attacks*. According to the botnet communication patterns [11], once a bot infection occurs, the bot registers itself to C2 and the keep-alive communication between the bot and C2 starts periodically. A botmaster issues a command to bots for launching an attack through C2, and then an attack is launched. The key idea in this study is that traffic patterns from or to C2 will be changed before the attack. For example, a botmaster would prefer more bots to launch an attack effectively, and then the bot register may increase before the attack. A command contains some data such as the target and parameter, and will be sent to numerous bots in parallel so that the size of the packet may be increased before the attack. Such predictive threat intelligence is crucial for an internet service provider (ISP) for prioritizing C2s and cutting off communication between C2 and bots in advance in crisis situations or at customers' requests.

This poster presents a method for effective measurement, labeling, and classification of botnet C2s for predicting attacks. In the measurement phase, various metrics of C2 are computed with flow data collected with a certain sampling rate by an ISP for network management. Next, a set of C2 metrics is labeled for a certain period based on 1) the gradient of spam volume and 2) the result of the correlation analysis between the moving average of each C2 metric and the volume of spam email associated with C2 as attack data. For the classification, a recurrent neural network is used to train and test the labeled dataset. The following sections describe each phase of the method and preliminary experimental results.

II. Methodology

A. Measurement

The 17 metrics listed in TABLE I are defined for measuring C2 activity, which is a compilation of various metrics studied in previous works [4], [10]. All metrics are computed with flow data for every three hours. Because the used flow data has a unidirectional format, IP addresses of C2 will be observed in the source or destination field. The metrics can be computed using three patterns: C2 in source, C2 in destination, and C2 in either source or destination. Finally, because the 17 metrics are multiplied by the three patterns, it is possible to use 51 metrics. The lists of C2s are retrieved from websites [1], [3] and provided by a reliable research institution once daily.

TABLE I. C2 Metrics

Category	Metrics
Size	1) # of bots
	2) # of bots observed multiple flows
Volume	3) Average, 4) standard deviation and 5) sum of bytes
	6) Average, 7) standard deviation and 8) sum of packets
Frequency	9) # of flows
	10) # of flows with few packets (less than three packets)
	11) # of flows with short duration (less than one s)
	12) # of flows with small bytes (less than 500 bytes)
Load	13) Average, 14) standard deviation and 15) sum of duration
Lifetime	16) # of days flow was observed in the last seven days
	17) # of days flow was continuously observed in the last seven days

B. Labeling

Spam reputation data [2] is retrieved as attack data once daily. It includes information such as the IP address of the spam sender and spam volume in the last day. The total spam volume associated with C2 can be added up by associating an IP address of C2 and an IP address of the spam sender in flow data. As a preliminary experiment, it was analyzed whether the metrics could be useful for predicting the increase in spam during a week. At that time, the moving average of each C2 metric was taken with a one-day time window. This time window represents how far in advance a sign can be detected. To align the number of data elements with the

C2 metric, spam volume was padded with the same value as the previous data. According to the result of the correlation analysis between each C2 metric and spam volume for each botnet, a different botnet tends to show different correlation and there are metrics with a high positive correlation. Based on this observation, it was decided to use all metrics and the following two criteria were set for labeling: 1) there is at least one metric with a high correlation greater than or equal to 0.3; 2) there is a positive gradient of the spam volume. With these criteria, C2 with the increasing spam for one week can be set as a *True* label. These steps are repeated for the entire period while shifting the starting point by one day.

C. Classification

Long Short-Term Memory (LSTM) [9] was applied for binary classification. LSTM is a recurrent neural network capable of learning long-term dependencies. To apply LSTM, each C2 metric is scaled between 0 and 1, and then the time series data of each C2 metric is laterally shifted. Finally, it becomes 2,856-dimensional data. Various models have been tried, but a model of stacked LSTMs was selected because of its high accuracy. The model has a layered structure comprising LSTM, dropout, LSTM, dropout, and dense. A dropout layer is used for the regularization that randomly sets some of the dimensions of the input vector to zero at each update during training time, which helps prevent overfitting. The dense layer represents matrix vector multiplication. The values in the matrix are the trainable parameters that get updated during backpropagation. The model is configured using binary cross-entropy for the loss function, Adam as the optimization algorithm, and Sigmoid as the activation function.

III. Evaluation

A. Dataset

TABLE II describes the dataset used in the evaluation, which was generated between August 3, 2019 and November 1, 2019 (90 days) and labeled as described in Section II-B with a different time window of the moving average. The number of true and false data is unbalanced, so false data are randomly sampled to align with the number of true data for experiments.

TABLE II. Dataset

Time Window (hours)	# of True	# of False	# of C2s
12	1,893	8,641	234
24	2,291	8,154	234
48	2,423	7,888	233

B. Experiment

The proposed method was evaluated with respect to accuracy as well as computational time. Using Python 3.6.9 with Keras 2.2.4 on top of TensorFlow 1.14.0, experiments were performed for training and testing with 10-fold cross-validation on the *Enki* which is the High Performance Computing cluster at the National Institute of Standards and Technology. TABLE III demonstrates the average results of the 10-fold cross-validation. The highest accuracy is 0.981, when the time window of the moving average is 48 and the number of hidden units is 100. By increasing the number of hidden units, both training time and testing time almost linearly increased as expected. The effect of the time window is not noticeable because all data might have been properly learned.

TABLE III. Results of binary classification

(batch=128, epoch=200, dropout=0.3, learning rate=0.01)

Time Window (hours)	Units	Train (s)	Test (s)	Accuracy	Recall	Precision
12	100	2,201.90	2.37	0.923	1.000	0.883
	200	3,186.01	3.71	0.929	0.956	0.918
	300	4,256.90	5.93	0.938	1.000	0.902
	400	5,566.59	9.39	0.898	0.994	0.859
	500	7,037.26	13.69	0.934	0.999	0.890
24	100	2,658.80	2.98	0.956	0.998	0.923
	200	3,817.19	4.20	0.926	1.000	0.881
	300	4,976.25	6.98	0.921	0.998	0.882
	400	6,730.17	11.15	0.966	1.000	0.939
	500	8,535.01	16.35	0.955	0.986	0.931
48	100	2,830.37	2.97	0.981	1.000	0.966
	200	4,050.56	4.56	0.881	1.000	0.829
	300	5,354.73	7.21	0.805	0.642	0.968
	400	7,138.42	11.92	0.918	0.963	0.897
	500	9,056.58	17.29	0.939	0.906	0.973

IV. Conclusions and future work

By focusing on C2 traffic, the proposed method managed to detect an increase in spam email by a botnet with an accuracy of 0.981. The next challenge is how far in advance to predict the increase in spam email. It is necessary to extract C2 communication appropriately because C2 metrics are computed including legitimate flow if a legitimate server is compromised and used as C2. It is also essential to accurately track a C2 IP address because C2 changes the IP address to avoid detection. Although the method can be further improved, the method would also be applicable to the prediction of attacks such as DDoS if the data that associates bots with C2 is available.

References

[1] abuse.ch SSLBL Snort / Suricata Botnet C2 IP Ruleset. [Online]. Available: https://sslbl.abuse.ch/blacklist/sslipblacklist.rules

[2] Email & Spam Data. [Online]. Available: https://talosintelligence.com/reputation_center/email_rep#spam-ip-senders

[3] Master Feed of known, active and non-sinkholed C&Cs IP addresses. [Online]. Available: https://osint.bambenekconsulting.com/feeds/c2-ipmasterlist-high.txt

[4] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Proceedings of the 2014 IEEE Conference on Communications and Network Security*, Oct 2014, pp. 247–255.

[5] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: Detecting Botnet Command and Control Servers Through Large-scale NetFlow Analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, 2012, pp. 129–138.

[6] W. Chang, A. Mohaisen, A. Wang, and S. Chen, "Measuring Botnets in the Wild: Some New Trends," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, 2015, pp. 645–650.

[7] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," in *Proceedings of the 15th Annual Network and Distributed System Security Symposium*, 2008.

[8] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, 2019.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics," *Computers & Electrical Engineering*, vol. 50, pp. 91–101, 2016.

[11] G. Vormayr, T. Zseby, and J. Fabini, "Botnet Communication Patterns," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2768–2796, 2017.

National Institute of Standards and Technology
U.S. Department of Commerce

Method for Effective Measurement, Labeling, and Classification of Botnet C2s for Predicting Attacks

Mitsuhiro Hatada and Matthew Scholl, Computer Security Division

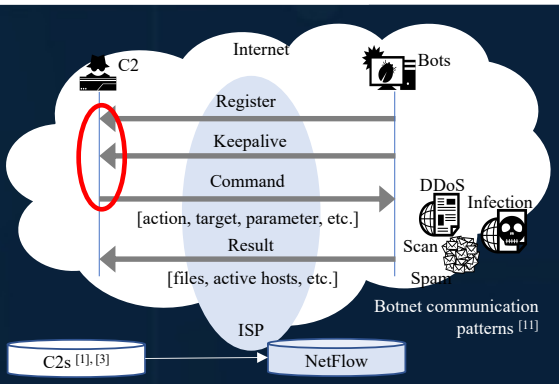
Motivation

- Botnet is still a serious threat to cybersecurity as it controls a massive number of compromised hosts to conduct various attacks such as sending email spam or launching a Distributed Denial-of-Service (DDoS) attack.
- Command and Control (C2) server plays a significant role in a botnet: it sends commands to bots and receives outputs of bots while hiding a botmaster behind it.
- Despite many previous attempts at botnet measurement [6], [8] and botnet detection [4], [5], [7], [10], to the best of our knowledge, there is no study that **detects the early signs of botnet attacks**.

Key Idea

Traffic patterns from or to C2 will be changed before the attack.

- Botmaster would prefer more bots to launch an attack effectively, and then the bot register may increase before the attack.
- Command contains some data such as the target and parameter, and will be sent to numerous bots in parallel so that the size of the packet may be increased before the attack.

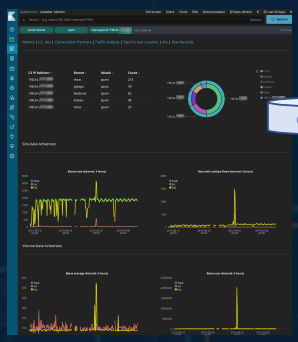


Measurement

- All metrics are computed with flow data for every three hours and can be computed using three patterns: C2 in source field, C2 in destination field, and C2 in either source or destination field of unidirectional NetFlow.
- Finally, it is possible to use 51 metrics.

TABLE I. C2 METRICS

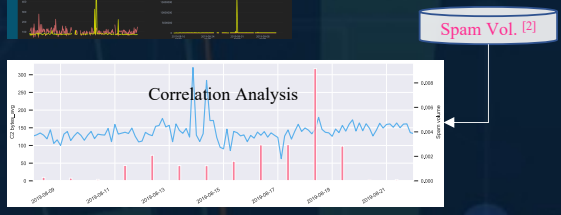
Category	Metrics
Size	1) # of bots
	2) # of bots observed multiple flows
Volume	3) Average, 4) standard deviation and 5) sum of bytes
	6) Average, 7) standard deviation and 8) sum of packets
Frequency	9) # of flows
	10) # of flows with few packets (less than three packets)
	11) # of flows with short duration (less than one s)
	12) # of flows with small bytes (less than 500 bytes)
Load	13) Average, 14) standard deviation and 15) sum of duration
Lifetime	16) # of days flow was observed in the last seven days
	17) # of days flow was continuously observed in the last seven days



Preliminary Analysis

Could the metrics be useful for predicting the increase in spam during a week?

- The total spam volume associated with C2 can be added up by associating an IP address of C2 and an IP address of the spam sender in flow data.
- The moving average of each C2 metric was taken with a one-day time window. This time window represents how far in advance a sign can be detected.
- According to the result of the correlation analysis between each C2 metric and spam volume for each botnet, *a different botnet tends to show different correlation* and *there are metrics with a high positive correlation*.



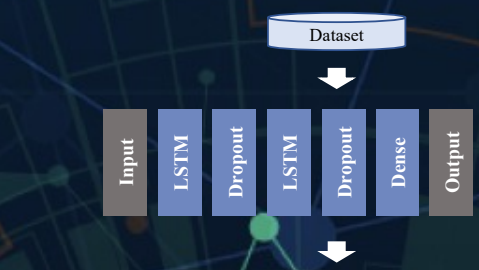
Labeling

- Criteria
  1) There is at least one metric with a high correlation greater than or equal to 0.3
  2) There is a positive gradient of the spam volume
- C2 with the increasing spam for one week can be set as a True label. These steps are repeated for the entire period while shifting the starting point by one day.

C2	Time	Metric A	Metric B	...	Label	2) Gradient
198.51.100.22	8/3 00:00-03:00	133.45	1872.10	...	True	0.02
	8/3 03:00-06:00	664.87	3606.19	...		
	:	:	:	...		
	8/9 21:00-24:00	293.42	18,341.22	...		
1) Correlation		0.11	0.42	...		

Classification

- Long Short-Term Memory (LSTM) [9], a recurrent neural network capable of learning long-term dependencies, was applied for binary classification.
- The model is configured using binary cross-entropy for the loss function, Adam as the optimization algorithm, and Sigmoid as the activation function.
- Each C2 metric is scaled between 0 and 1, and then the time series data of each C2 metric is laterally shifted. Finally, it becomes 2,856-dimensional data.



Evaluation

- Dataset was generated between August 3, 2019 and November 1, 2019 (90 days) with a different time window of the moving average.
- The number of true and false data is unbalanced, so false data are randomly sampled to align with the number of true data for experiments.
- Experiments were performed for training and testing with 10-fold cross-validation on the Enki which is the High Performance Computing cluster at the NIST.
- By focusing on C2 traffic and using LSTM for the time series data of comprehensive metrics, the proposed method managed to detect an increase in spam email by a botnet with an accuracy of 0.981.

TABLE II. DATASET

Time Window (hours)	# of True	# of False	# of C2s
12	1,893	8,641	234
24	2,291	8,154	234
48	2,423	7,888	233

TABLE III. RESULTS OF BINARY CLASSIFICATION

(batch=128, epoch=200, dropout=0.3, learning rate=0.01)

Time Window (hours)	Units	Train (s)	Test (s)	Accuracy	Recall	Precision
12	100	2,201.90	2.37	0.923	1.000	0.883
	200	3,186.01	3.71	0.929	0.956	0.918
	300	4,256.90	5.93	0.938	1.000	0.902
	400	5,566.59	9.39	0.898	0.994	0.859
	500	7,037.26	13.69	0.934	0.999	0.890
24	100	2,658.80	2.98	0.956	0.998	0.923
	200	3,817.19	4.20	0.926	1.000	0.881
	300	4,976.25	6.98	0.921	0.998	0.882
	400	6,730.17	11.15	0.966	1.000	0.939
	500	8,535.01	16.35	0.955	0.986	0.931
48	100	2,830.37	2.97	0.981	1.000	0.966
	200	4,050.56	4.56	0.881	1.000	0.829
	300	5,354.73	7.21	0.805	0.642	0.968
	400	7,138.42	11.92	0.918	0.963	0.897
	500	9,056.58	17.29	0.939	0.906	0.973

References

[1] abuse.ch SSLBL Snort / Suricata Botnet C2 IP Ruleset. [Online]. Available: https://sslbl.abuse.ch/blacklist/sslipblacklist.rules
[2] Email & Spam Data. [Online]. Available: https://talosintelligence.com/reputation_center/email_rep#spam-ip-senders
[3] Master Feed of known, active and non-sinkholed C&Cs IP addresses. [Online]. Available: https://osint.bambenekconsulting.com/feeds/c2-ipmasterlist-high.txt
[4] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in Proceedings of the 2014 IEEE Conference on Communications and Network Security, Oct 2014, pp. 247–255.
[5] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: Detecting Botnet Command and Control Servers Through Large-scale NetFlow Analysis," in Proceedings of the 28th Annual Computer Security Applications Conference, 2012, pp. 129–138.
[6] W. Chang, A. Mohaisen, A. Wang, and S. Chen, "Measuring Botnets in the Wild: Some New Trends," in Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, 2015, pp. 645–650.
[7] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," in Proceedings of the 15th Annual Network and Distributed System Security Symposium, 2008.
[8] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet," in Proceedings of the 26th Annual Network and Distributed System Security Symposium, 2019.
[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics," Computers & Electrical Engineering, vol. 50, pp. 91–101, 2016.
[11] G. Vormayr, T. Zseby, and J. Fabini, "Botnet Communication Patterns," IEEE Communications Surveys Tutorials, vol. 19, no. 4, pp. 2768–2796, 2017.