IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 25, No. 1, February, 2020, pp. 127-130
Journal homepage: http://iieta.org/journals/isi

Android Botnet Detection Using Machine Learning

Mohammad M. Rasheed[1*], Alaa K. Faieq[2], Ahmed A. Hashim[1]

[1] College of Engineering, University of Information Technology and Communications, Baghdad 10013, Iraq
[2] Computer Technology Engineering Department, Baghdad College of Economic Sciences University, Baghdad 10090, Iraq

Corresponding Author Email: mohammad.rasheed@uoitc.edu.iq

https://doi.org/10.18280/isi. 250117

Received: 7 November 2019
Accepted: 10 January 2020

Keywords:
mobile security, botnet detection, machine learning detection

ABSTRACT

A botnet is a network of agreed nodes spreading malware software, usually installed by all varieties of attacking methods likes worms, Trojan horses, and viruses. Many techniques have recently been proposed to block mobile malware or detect it. But our model is different on another that proposed before, it focused on 81 attributes that collected from network traffic features. We tested ten of android botnet, which are Beanbot, Biige, Fakeinst, FakeMart, FakeNotify, Jifake, Mazarbot, Nandrobox, Plankton, and SMSsniffer using Weka machine learning. We have 32762 instances, which classified as attack and not attack. We used WEKA machine learning and we tested SMO, Random Tree, J48, Naïve Bayes and LMT algorithms. The best result to classify the botnet attack was 85%. The contribution of this paper is detected major of android botnet in different scenario because we are using 81 attributes. In future work, we will attach new sub algorithm in machine learning, to improve accuracy of the result of detecting more mobile malware.

1. INTRODUCTION

The botnet's etymological concept is derived from the term "bot", which means that the victim is controlled by an attacker. The use of botnets has increased dramatically recently. Botnets are a number of computers that connected to the internet have the large accumulative bandwidth and computing power. The attacker, also known as botmaster, can control large networks of botnet from different locations to launch attacks. Botnet is characterized by distributed denial of service attacks, email spam, key logging and also password cracking. Currently, botnets are the one of the greatest threats to the Internet [1].

There are various element techniques of botnets that make them almost unique in the structure, by capabilities and technical implementation. However, there are always a botmaster, one or more command and control servers, and at least one but usually thousands controlled nodes. A botnet is like internet worm that are infected nodes, which is executing commands while trying to stay hidden from anti-malware to detect it. The botmaster is the master of the all Botnet. In the state of attack the service. Only a part of the botnet nodes can usually be partially managed by a client. The instruction set available to the client is usually a subset of the entire instruction set. In fact, anyone who controls the botnet at any given time is the real attacker [2].

The general model followed by most botnet Prevailing botnet Command and Control channels is comprised of three steps: The first step, the botmaster must send an instruction command to the botnet. The second step, the botnet must be responded to the command by carrying out activities; and final stage, the results of the activities are sent back to the botmaster by the botnet. There are three techniques of command and control channels nodes: (a) HTTP based command and control channels that use a pull-based model in which botnets regularly poll the command and control server to ask new commands. (b) Internet Relay Chat (IRC) based command and control channels that used a push based method in which the botmaster sends new instruction commands to the botnet, which then responds immediately to the commands, (c) peer to peer (P2P) based command and control channels, in which P2P communication is applied to proxy commands or to find a command and control server. P2P-based command and control has the advantage that there is no single point of failure that is unique to HTTP-based botnet and IRC-based [3, 4].

Most of the existing detection techniques can only detect malware Android applications, however, Android botnet applications cannot be detected [2], so the article focused on the detection of botnet Android applications. The remainder of this paper is organized as follows: Section 2 provides a brief description of related work; Section 3 deals with the System approach of android botnet detection using machine learning; Section 4 shows the results of our system approach; and Section 5 is the conclusion and the recommendations for future research.

2. RELATED WORK

The two main methods of detecting malware can be broadly categorized as anomaly, and misuse detection methods [5-17]. The methods of detecting damage can be classified into two type's methods, detection that uses common features of the malware applications and methods that are based on signatures or models to detect of known malware [13, 14]. To detect the malware depend on "generic feature" is very limited to anti malware application. The research challenge is not easy to create the algorithm for detecting unknown malware functions is very hard. Consequently, such methods are very limited to a limited class of recognized the malware. Signature-based methods have another way to detect the malware but the disadvantage of this technique depends on the signature models that are applied before and cannot be used for zero-day malware detection. Anomaly-based detection technique [15, 16] need models with certain functionality to be specified, which is even more complex since they need a much broader function's code for coverage compared to the detection of malware functions. Botnet detection methods can be divided into two categories: Network-based detection (NBD) [18] and host-based detection (HBD) [19].

2.1 Host based detection (HBD)

HBD is the most advanced technique. This technique is work to decide if the host at a risk, this technique constantly observes changes in processes of network connections, files and registrations in a controlled environment [20, 21]. HBD is beneficial in detecting known bots. However, its performance is poor because new or different botnet cannot be recognized. For example, HBD is unable to detect botnet with new model such as a counter debugger and a rootkit.

Some of HBD techniques look into the contents of file system to detect botnet. Because the botnets are binary executable and exist within the system's file, where a file signature is compared with the file binary to looking for botnet signatures on the file system. This is a popular technique that used to look for botnets [22].

2.2 Network-based detection (NBD)

Sarnsuwan et al. [23] proposed a method to detect the malware by using data mining, where it involves the use of data analysis way to discover unknown knowledge by valid relationships and patterns in large data sets. These tools can include, mathematical algorithms, statistical models and machine learning methods. So that, data mining comprises of more than collecting and managing data. It also includes analysis after that can be predicted. Sarnsuwan et al. [22] used three data mining algorithms that are C4.5 Decision Tree, Random Forest, and Bayesian network.

NBD [24-26] essentially knows the traffic network in the command and control phrase of every botnet, because the behavioral characteristics is different from phrase to another phrase. NBD focuses primarily on examining two types of network behavior: the rate of failure connections and flow features. The algorithms that depend on used flow features that include the number of uplink and downlink of data packets, the average length of uplink and downlink of data packets, the number of uplink and downlink, transmission bytes, the duration time of data flow, the maximum length of downlink and uplink of data packets, the total length of loaded data packets in a flow, the rate of the length of data packets in uplink and downlink, and the average length of downlink and uplink of data packets.

Currently, researchers are adding neural network and machine learning to NBD to detect unknown botnet traffic network. Moreover, this technique is a hot research point in the analysis of botnet traffic and detection [27].

The NBD technique has a high detection rate due to the fact that common flow characteristics are extracted regardless of the botnet types. Nevertheless, in the high speed and complex network, existing detection platforms that are based on flow characteristics are useless due to the high packet drop rate.

In this article used NBD methods that try to detect botnet that infected the devices by correlating the similar of network traffic among different mobile devices using monitored network that collected by Lashkari et al. [28]. We used NBD technique and we add machine learning algorithm to detect botnet, so that our technique does not require of any prior knowledge of botnet signatures.

3. SYSTEM APPROACH

Lashkari et al. [28] run the malware and harmless applications on real smartphones to avoid changing the runtime behavior of advanced malware samples that can detect the emulator environment. To get a comprehensive overview of our malware samples. Lashkari et al. [28] have created a specific scenario for each malware category. The system approach also defined three states of data collection to overcome the stealth of advanced malware. The system approach consisted of three stages: The first is the installation: The first data collection status, which takes place immediately after the malware is installed (1-3 minutes). The second is before the restart: the second data collection status, which occurs 15 minutes before the phone restarts. The third is after the restart: The last data collection status that occurs 15 minutes after the phone restarts.

For the function extraction and selection, the system recorded network traffic functions (.pcap files) and extracted over 80 functions using CICFlowMeter-V3 in all three states mentioned (installation, before restart and after restart). Lashkari et al. [28] collected the data by generating bidirectional flows of packets between the source and destination, therefore it is included 81 statistical network traffic features such as Length of packets, Number of packets, Number of bytes, and Duration.

The output of CICFlowMeter-V3 is a CSV format file that is as columns such as (Destination IP, Source Port, Protocol, Destination Port, Source IP and Flow ID,) with 75 network traffic features. The TCP protocol flows packets are normally terminated when the connection is closed by FIN packet, but UDP protocol flows packets are terminated by a flow timeout. Therefore, the flow timeout value can be arbitrarily assigned by the individual scheme, e.g., ten minutes for both UDP and TCP.

4. RESULT

Machine learning is a type of artificial intelligence which used the relationships between data and information through the systematic algorithms. Machine learning systems can be trained the algorithm depend on historical information to build recognition systems such as the iPhone 'Siri' to convert audio information from a sequence of speech data into semantic structures, which are expressed in the form of a word sequence. Machine learning is public uses in web search engine, stock market prediction, weather forecasting, ad placement, gene sequence analysis, drug development, behavior analysis, credit scoring, behavior analysis, big data analytics, smart coupons, and a variety of other applications. A Classification is a technique which received a new input, but it is unlabeled feature, or an instance of observation and identifies a class or category depend on training. As shown below in Figure 1. Usually this technique used classifiers that use statistical inference (a measure of probability) to categorize the best term for a particular instance [29].

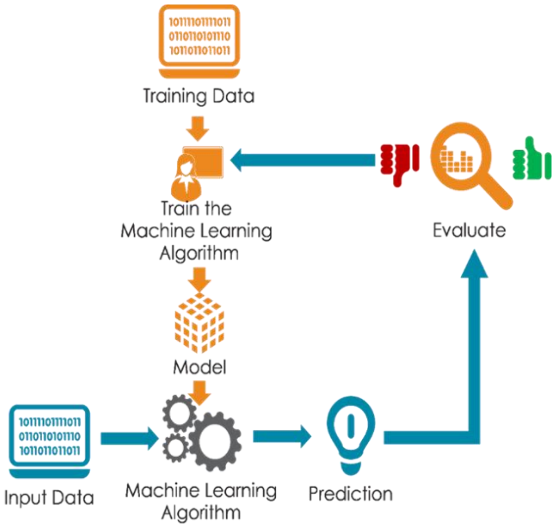
Training Data
Train the Machine Learning Algorithm
Evaluate
Model
Input Data
Machine Learning Algorithm
Prediction

Figure 1. Machine learning algorithm

We tested ten datasets of android botnet, which are Beanbot, Biige, Fakeinst, FakeMart, FakeNotify, Jifake, Mazarbot, Nandrobox, Plankton, and SMSsniffer using Weka machine learning. We have 32762 instances and 85 attributes that create by Lashkari et al. [28]. We used 66% instance in training and 34% in testing. The result for Naive Bayes and J48 as shown in below table. But we show there is a relation between some attribute and test environment. We think this test is not correct as shown in Table 1.

Table 1. The first test

Algorithm Name	Correctly Classified	Incorrectly Classified
NaiveBayes	72.39%	27.61%
J48	95.35%	4.65%

So that we removed (flow ID, Source IP, Destination IP, Time Stamp), for example the ‘flow ID’ and ‘time stamp’ are related to test in malware and benign samples. But in real attack there are malware and benign sample at the same time. The data are collected by spread malware and benign, so “flow ID and Time Stamp” are sequenced in both in malware and benign sample, so that any classification method can find this relation sequence, and this happened because we spread the environment test. So that we must remove it. Furthermore, source and destination IP must be removed because is related with infector and the victim, and these will be changed in real test, because the malware is attacking different IP and the infector IP will be changed also. So that these tests are more correct and the attribute we selected is not attach to test environment. We depend on 81 attribute that captured from the network, after remove four attribute and 32762 instances. The result shows J48 is more than 85% correct classification as shown in Table 2.

Table 2. The second test

Algorithm Name	Correctly Classified	Incorrectly Classified
SMO	72.47	27.53
Random Tree	80.25	19.75
J48	85.45	14.55
Naïve Bayes	71.50	28.50
LMT	84.37	15.63

5. CONCLUSION AND FUTURE WORK

The sample that used in our tests is classified as ‘Attack’ and ‘Not attack’. The sample ‘Not Attack’ is clean, that meaning every data doesn't include any attack. But in ‘Attack’ sample is not included only attack because the botnet malware when work on the network, there are a lot of normal network programs that work with botnet, but we capture the group of packets and we classify it as “Attack”. We show the result is needed to improve, why? Because best result is 85%. So that in future work need to improve the algorithm classification by attaching new sub algorithm to machine learning.

REFERENCES

[1] Wang, P., Wu, L., Aslam, B., Zou, C. (2015). Analysis of Peer-to-Peer Botnet Attacks and Defenses. In: Król, D., Fay, D., Gabryś, B. (eds) Springer, Berlin, 183-214. https://doi.org/10.1007/978-3-319-15916-4_8

[2] Kirubavathi, G., Anitha, R. (2018). Structural analysis and detection of android botnets using machine learning techniques. International Journal of Information Security, 17(2): 153-167. https://doi.org/10.1007/s10207-017-0363-3

[3] Fedynyshyn, G., Chuah, C., Tan, G. (2011). Detection and Classification of Different Botnet C&C Channels. In: Calero, A., Yang, T., Mármo, G., García, J., Li, X., Wang, Y. (eds), Autonomic and Trusted Computing. Springer, Berlin, 228-242. https://doi.org/10.1007/978-3-642-23496-5_17

[4] Bederna, Z., Szadeczky, T. (2020). Cyber espionage through Botnets. Security Journal, 33: 43-62. https://doi.org/10.1057/s41284-019-00194-6

[5] Rasheed, M., Norwawi, N., Ghazali, O., Faaeq, M. (2019). Detection algorithm for internet worms scanning that used user datagram protocol. International Journal of Information and Computer Security, 11(1): 17-32. https://doi.org/10.1504/IJICS.2019.096847

[6] Rasheed, M., Faaeq, M. (2019). Behavioral detection of scanning worm in cyber defense. In: Arai K., Bhatia R., Kapoor S. (eds) Proceedings of the Future Technologies Conference (FTC), pp 214-225. https://doi.org/10.1007/978-3-030-02683-7_16

[7] Rasheed, M., Badrawi, S., Faaeq, M., Faieq, A. (2017). Detecting and optimizing internet worm traffic signature. Proceedings of the 8th International Conference on Information Technology, pp. 870-874. https://doi.org/10.1109/ICITECH.2017.8079961

[8] Rasheed, M., Ghazali, O., Budiarto, R. (2012). Fast detection of stealth and slow scanning worms in transmission control protocol. Journal of Applied Sciences, 12: 2156-2163. http://doi.org/10.3923/jas.2012.2156.2163

[9] Rasheed, M. Kadhum, M. (2008). Traffic signature detection for unknown internet worms. In IEEE International Conference on Network Applications, Protocols and Services, 2008, Malaysia, pp. 1-5.

[10] Rasheed, M., Ghazali, O., Norwawi, N., Kadhum, M. (2009). A traffic signture-based algorithm for detecting scanning internet worms. International Journal of Communication Networks and Information Security, 1: 24-30.

[11] Rasheed, M., Ghazali, O., Norwawi, N. (2010). Server scanning worm detection by using intelligent failure connection algorithm. Research Journal of Information Technology, 2: 228-234. http://doi.org/10.3923/rjit.2010.228.234

[12] Rasheed, M., Ghazali, O., Norwawi, N. (2012). Intelligent signature detection for scanning internet worms. Information Technology Journal, 11: 260-267. http://doi.org/10.3923/itj.2012.760.767

[13] Egele, M., Kruegel, C., Kirda, E., Vigna, G. (2011). PiOS: Detecting privacy leaks in iOS applications. Proceedings of the 18th Annual Network and Distributed System Security Symposium, pp. 177-183.

[14] Enck, W., Gilbert, P., Chun, B., Cox, L., Jung, J., McDaniel, P., Sheth, A. (2010). TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, USENIX Association, pp. 1-6.

[15] Shabtai, A., Tenenboim, L., Mimran, D., Rokach, L., Shapira, B., Elovici, Y. (2014). Mobile malware detection through analysis of deviations in application network behavior. Computers & Security, 43: 1-18. https://doi.org/10.1016/j.cose.2014.02.009

[16] Kim, H., Smith, J., Shin, K. (2008). Detecting energy-greedy anomalies and mobile malware variants. In Proceedings of the 6th International Conference on Mobile Systems. Applications, and Services, pp. 239-252. https://doi.org/10.1145/1378600.1378627

[17] Rostami, M., Shanmugam, B., Idris, N. (2011). Analysis and detection of P2P botnet connections based on node behavior. Proceedings of the World Congress on Information and Communication Technologies, pp. 928-933. https://doi.org/10.1109/WICT.2011.6141372

[18] Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J., Hakimian, P. (2011). Detecting P2P botnets through network behavior analysis and machine learning. In Proceedings of the 9th Annual International Conference on Privacy, Security and Trust (PST '11), Canada, pp. 174-180. https://doi.org/10.1109/PST.2011.5971980

[19] Zhang, H., Gharaibeh, M., Thanasoulas, S., Papadopoulos, C. (2016). Botdigger: detecting DGA bots in a single network. Proceedings of the IEEE International Workshop on Traffic Monitoring and Analaysis, Belgium, pp. 1-8.

[20] Technical report, BotDigger: Detecting DGA Bots in a Single Network, CS-16-101, available at http://www.cs.colostate.edu/~hanzhang/papers/BotDigger-techReport.pdf.

[21] Wang, W. Fang, B., Cui, X. (2010). Botnet detecting method based on group-signature filter. Journal on Communications, 31(2): 29-35.

[22] Daniel J., Sanok, J. (2005). An analysis of how antivirus methodologies are utilized in protecting computers from malicious code. Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, Georgia, 2005, pp. 142-144. https://doi.org/10.1145/1107622.1107655

[23] Sarnsuwan, N. Charnsripinyo, C., Wattanapongsakorn, N. (2010). A new approach for internet worm detection and classification. In 6th International Conference on Networked Computing, pp. 1-4.

[24] Shanthi, K., Seenivasan, D. (2015). Detection of botnet by analyzing network traffic flow characteristics using open source tools. In Proceedings of the 9th IEEE International Conference on Intelligent Systems and Control (ISCO '15), India, pp. 1-5. https://doi.org/10.1109/ISCO.2015.7282353

[25] Kirubavathi, G., Anitha, R. (2016). Botnet detection via mining of traffic flow characteristics. Computers and Electrical Engineering, 50: 91-101. https://doi.org/10.1016/j.compeleceng.2016.01.012

[26] Zhang, J., Perdisci, R., Lee, W., Luo, X., Sarfraz, U. (2014). Building a scalable system for stealthy P2P-botnet detection. IEEE Transactions on Information Forensics and Security, 9(1): 27-38. https://doi.org/10.1109/TIFS.2013.2290197

[27] Chen, R., Niu, W., Zhang, X., Zhuo, Z., Lv, F. (2017). An Effective conversation-based botnet detection method. Mathematical Problems in Engineering, 2017: 1-9. https://doi.org/10.1155/2017/4934082

[28] Lashkari, A., Draper-Gil, G., Mamun, M., Ghorbani, A. (2017). Characterization of tor traffic using time based features. In the proceeding of the 3rd International Conference on Information System Security and Privacy, Portugal, pp. 253-262. https://doi.org/10.5220/0006105602530262

[29] Awad, M., Khanna R. (2015). Machine learning. In: Efficient Learning Machines. Apress, Berkeley, CA, pp. 1-18. https://doi.org/10.1007/978-1-4302-5990-9_1