Detection of HTTP Based Botnets

With Network Analyzers using Classifiers and Domain Name Filters

Ravi Kumar Reddy K [1], Anantha Krishna V [2], Dr Anitha J [3]
[1,3] Dept. of Computer Science and [2] Dept. of Electrical and Electronics Engineering,
DSATM, Bangalore

Abstract— **Botnets have been one of the noxious threats to cyber security. They have been the cause for several Data breaches and illegal online money generation in the past decade. Botnets need a Command and Control(C&C) mechanism for communication among all the bots. HTTP based botnets use the HTTP protocol to publish commands on certain web servers. Recently, Botnet detection systems such as SVM based Domain filtering and C4.5 and Naïve Bayes based network analyzers have been proposed. However, the delay time and efficiency of these systems have proved to be low comparatively. To overcome these lags, a system where machine learning classifiers such as See5, Naïve Bayes and domain name classifications using SVM Lite are used to detect HTTP based botnets in a more responsive, quick and efficient manner, is proposed in this work. These new techniques are implemented by classifying the incoming traffic of a network into flows using a network analyzer and collector such as ntop and nProbe. The proposed system obtained very promising performance on detecting HTTP based Botnets.**

Keywords—Botnets, Botnet Detection, DNS filtering.

I. INTRODUCTION

A bot is an application that can perform certain task redundantly. Collection of bots with an established communication system is a botnet. The individual who is responsible for and maintains a bot is a botmaster. Botnets need a rudimentary command control system for communication. Over the years, IRC (Internet Relay Chat), P2P system, and HTTP based systems have been used for this purpose. Botmasters have begun to use the central C&C model again, where the HTTP protocol is used to publish the commands on certain web servers. Instead of remaining in connected mode, the HTTP bots periodically visit certain web servers to get updates on new commands. This model is called the PULL style and continues at a regular interval that is defined by the botmaster. Botmasters use HTTP protocol to hide their activities among the normal web flows and easily avoid current detection methods like firewalls. Most dangerous bots that were extremely chaotic and their effects have been enlisted in [8].

Botnets basically rely on the network topology. The first generation botnets used Internet Relay Chat for communication. The IRC bots follow the PUSH approach as they connect to selected channels and remain connected. In response to this, Peer to Peer Protocols were put forward by the Botmasters in which a command sent to a node is forwarded to its neighbors by corresponding nodes. But this approach proved difficult as commands were not completely under control. As a result, HTTP based bots came into use where the HTTP protocol is used to publish commands on certain webservers. This resulted in major data breaches in several critical systems leading to major security loopholes of availability in cyber security.

Botnet detection system has become a matter of critical concern in the computing industry. Many techniques have been put forward for detecting botnets. IP flow system is one the most adopted systems. They use C4.5 and Naïve Bayes algorithms as classifiers which are slow (about 0.11 – 0.16 sec for major bot detections [2]) and highly memory consuming (about 90% more in case of forest dataset [9]) and have shown a lag in efficiency by displaying extreme changes for smaller changes in parameters [9]. In response to these flaws, this work uses the See5 Algorithm. It forms faster and smaller Decision Trees and its rule sets occupy less memory comparatively. They are faster, accurate and highly optimized [9]. The basic comparison statistics between Decision tree algorithms and Naïve Bayes algorithm on various parameters is provided in Table 1.1

Further, for generating flow based traffic, NetFlow tools are currently in use. These are blind to local traffic and their visibility is limited to routed traffic. Moreover NetFlow overhead can overtax infrastructure. In the proposed system, ntop and nProb have been used as flow based network exporting and collecting tools as they are applicable to ubiquitous networks and are flexible comparatively[1]

II RELATED WORK

Fariba Haddadi et al. developed a system for Botnet Behaviour analysis using IP Flows [1]. In this work, a botnet detection system is developed using NetFlow tool (Ex:Softflowd) to classify network traffic into flows and legitimate traffic is divided based on C4.5 and Naïve Bayes Algorithm. Strayer et al. developed an IRC botnet detection framework that makes use of machine learning techniques [2]. A three layer approach has been used in this work, a clustering methodology to identify activities, a classification methodology to filter traffic and a topology analyzer to detect botnets. Kirubavathi et al. designed HTTP based botnet detection system using a multilayer Feed-Forward Neural network [3]. As web-based botnets periodically make a web request from the C&C web server to download the instructions, they extract features related to TCP connections in specific time intervals. These features are used to detect them. Zeidanloo et al. proposed a detection system focusing on P2P and IRC based botnets [4]. In this work, similar three

layer approach has been implemented to differentiate between different legal botnets in a network. Haddadi et al. proposed Stateful-SBB system, to detect automatically generated malicious (botnet) domain names [5]. Stateful-SBB could differentiate botnet C&C domain names, which are located in the network packet payload. Francois et al. proposed a NetFlow monitoring framework that leveraged a simple host dependency model to track communication patterns and employed linkage analysis and clustering techniques to identify similar botnet behavioral patterns [6].

In the proposed work, ntop and nProb are employed for extracting network traffic into IP Flows, a DNS based domain name filtering technique for initial filtering of malicious domain names and two machine learning techniques utilizing the extracted features to detect botnet behavior. The complete system ensures an equivocal, comparatively efficient and quick, two-level botnet detection system.

Table 1.1: Comparison of Classifiers

Parameters	**Decision Tree Algorithms**	**Naïve Bayes Algorithm**
Accuracy in General	2	1
Speed of Learning with respect to number of attributes and number of instances	3	4
Speed of Classification	4	4
Tolerance to missing Values	3	4
Tolerance irrelevant attributes	3	2
Tolerance to redundant attributes3	2	1
Dealing with Discrete/Binary/Continuous Attributes	4	3(not continuous)
Tolerance to noise	2	3
Dealing with over fitting	2	3
Attempts of incremental Learning	2	4
Exponential ability /transparency of knowledge/classifications	4	4

III METHODOLOGY

In this work, we are detecting botnets using two prolific machine learning classifiers that offer a rudimentary encryption level. Therefore, we do not have the access to the payload packet. Thus, we find out the possibility of detecting botnets by aggregating the traces of the network into flows using the flow exporters, ntop and nProb. Further, we study the potential futures chosen by the classifiers from a given set. In order to achieve these, we built the following modules for the proposed system. (a) Network Exportation module (b) Network Collection module, and (c) Network Traffic classification module

A. Network Exportation

The essence of this work is to identify botnet behaviour using domain fluxing techniques as botnets follow fluxing techniques as their strength point in their latest versions. Even though these types of botnets are the most frequent on the operation in the field lately, as there are no publically available traffic publically available lists of c&c domain names are employed in order to generate significant representative traffic.

1) *Boatnet:* Boatnet entered the world of internet in the year 2013 and had captive of more than 500+ server computers and had a spam capacity of about 0.01 bn/day [8]. There is another botnet with the name YOLOBotnet [8] which operates in a similar fashion as that of Boatnet The variant carried out attacks using the P2P network architecture targeting users of Facebook, Hotmail and Yahoo and Google

2) Zeus: Zeus malware responsible of a series of attacks against principal internet service providers. The variant carried out attacks using the P2P network architecture targeting users of Facebook, Hotmail and Yahoo and Google Mail. The Zeus is one of the most notorious malware that we have found in several cases. We can consider it as one of the best products of the malware industry. The malware is really appreciated by cyber criminals that have improved its feature over the months. Zeus is born as an agent able to steal banking information by logging keystrokes and form grabbing; then it is spread mainly through phishing and driven-by download schemes [10].

3) Citadel: In May 2011, source code for the infamous Zeus Trojan horse was leaked on the Internet. In addition to providing a glimpse inside a notorious piece of adversarial tradecraft, the source code provided an opportunity for enterprising malware authors to meet an emerging demand for cybercrime tools. Two major toolkits based on the leaked Zeus source code have become renown in the marketplace: ICE IX and Citadel. In this work, we obtained the list of citadel botnet command and control centres names from the citadel botnet section of the Zeus tracker [11].

Many other Botnets have been used for analysis as there were many destructive botnets which are similar to Zeus and Citadel such as Conficker, Ramnit etc.

B. Network Collection

Flow generation tools summarise the whole traffic which utilizes the packet headers in the network. These tools collect all the packets information with a similar parameters such as IP addresses and port numbers, combine them into flows and then estimate statistics such as net flow i.e. the number of packets per flow etc. Cisco Systems NetFlow Services clearly

states flow as "a unidirectional sequence of packets with some common properties that pass through a network device" in RFC 3954. The very most common way of finding the IP flow is by combining all the five properties i.e. 1)Source/Destination IP addresses 2) Source/Destination port numbers, and a Protocol for the traversal.

C. *Network Traffic Classification*

See5, one of the most preferred classification algorithm, and Naïve Bayes, preferred for its simplicity and optimization, are used in this work as Machine Learning Classifiers.

a) See5.0 Algorithm: See5.0 is a decision tree algorithm, which Quinlan is commercially selling (single-threaded version is distributed under the terms of the GNU). The advantages of See5 are that its several orders of magnitude faster, memory efficienct and forms very accurate and smaller decision trees. It has the ability to weight different attributes, and winnowing (reducing noise). During the experiment we used the differentiation attribute sets. To get a clear perspective on impact of HTTP filters on network traffic, normal ordinary decision-tree based classification for two different cases: for HTTP traffic only and other for the whole traffic have been performed.

The classifier is trained and tested first. Then the resulting decision tree or rule set is used to classify unseen data. See5.0 algorithm has many features like:

a. See5.0 algorithm can respond on noise and missing data.
b. See5.0 provides boosting.
c. A large decision tree may be difficult to read and comprehend.
d. See5.0 provides the option of viewing the large decision tree as a set of rules which is easy to understand.
e. Overfitting is solved by the See5.0 and Reduce error pruning technique.
f. See5.0 can also predict which attributes are relevant in classification and which are not. This technique, known as Winnowing is especially useful while dealing with high dimensional datasets. The Algorithm is presented below[7]:

Algorithm See5
Input: Example, Target Attribute, Attribute
Output: Decision tree that Classifies the data to the requisite conditions
Algorithm:
a. Check for the base class
b. Construct a DT using training data
c. Find the attribute with the highest info gain (A_Best)For each ti ϵ D, apply the DT to determine its classsince the application of a given tuple to a DT isrelatively straightforward.

b) Naïve Bayes Algorithm: A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. It assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Th mathematical approach of this algorithm is presented below:

Using Bayes Theroem we write,

$$p(C|F_1,\dots,F_n) = \frac{p(C)p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}$$

In plain English the above equation can be written as

$$posterior = \frac{prior \times likelihood}{evidence}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the join probability model.

$$p(C,F_1,F_2,\dots,F_n)$$

which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$p(C,F_1,F_2,\dots,F_n)$$
$$= p(C)p(F_1,F_2,\dots,F_n|C)$$
$$= p(C)p(F_1|C)p(F_2,\dots,F_n|C,F_1)$$
$$= p(C)p(F_1|C)p(F_2|C)p(F_3,\dots,F_n|C,F_1,F_2)$$
$$= p(C)p(F_1|C)p(F_2|C)\dots p(F_n|C,F_1,F_2,\dots,F_{n-1})$$

Now the "naive" conditional independence assumptions come into play. Assume that each feature F_i is conditionally independent of every other feature for. This means that for $j \neq i$, and so the joint model can be expressed as

$$p(F_i|C,F_j) = p(F_i|C)$$

For $j \neq i$, and so the joint model can be expressed as

$$p(C,F_1,F_2,\dots,F_n)$$

$$= p(C)p(F_1|C)p(F_2|C)\dots p(F_n|C,F_1,F_2,\dots,F_{n-1}) =$$

$$p(C) = \prod_{i=1} p(F_i|C)$$

This means that under the above independence assumptions, the conditional distribution over the class variable **C** can be expressed like this:

$$p(C|F_1,\dots,F_n) = \frac{1}{Z}p(C)\prod_{i=1} p(F_i|C)$$

where **Z** (the evidence) is a scaling factor dependent only on F1.....Fn, i.e., a constant if the values of the feature variables are known.

IV CONCLUSION

With the advent of Citadel, Conficker and Zeus, the major devastating effects of botnets have become a major threat against cyber security. In this work, two popular machine learning techniques are validated on their effects with some of the most noxious botnets noticed. The incoming network features are divided into flows using flow exporters and collectors and their performances are analyzed. The major influence of machine learning classifiers and domain name filtering systems on malicious botnet affected networks have been analyzed in this work.

Future Work will follow studying on different ML Classifiers for the same purpose and various filtering strategies employed in eradicating illegitimate botnets.

REFERENCES

[1]Fariba Haddadi, Jillian Morgan, Eduardo Gomes Filho, A. Nur Zincir-Heywood, "Botnet Behaviour Analysis using IP Flows with HTTP Filters using Classifiers" in 28th International Conference on Advanced Information Networking and Applications Workshops, 2014.

[2] W.T. Strayer, D. Lapsely, R. Walsh, and C. Livadas, "Botnet detection based on network behavior," *Advances in Information Security*, vol. 36, 2008.

[3] V. Kirubavathi and R.A. Nadarajan, "HTTP botnet detection using adaptive learning rate multilayer feed-forward neural network," in Information Security Theory and Practice: security, privacy and trust in computing systems and ambient intelligent ecosystems, 2012.

[4] H.R. Zeidanloo, A. Bt Manaf, P. Vahdani, F. Tabatabaei, and M. Zamani, "Botnet detection based on traffic monitoring," in Networking and Information Technology (ICNIT) , 2010.

[5] F. Haddadi and A.N. Zincir-Heywood, "Analyzing string format-based classifiers for botnet detection: GP and SVM," in IEEE Congress on Evolutionary Computation (CEC), 2013.

[6] J. Francois, Sh. Wang, R. State, and Th. Engel, "BotTrack: tracking botnets using NetFlow and PageRank," *Networking*, 2011.

[7] A.S Galathiya, A.P. Ganatra, C.K. Bhensdadia, “Improved Decision Tree Induction Algorithm with Feature Selectiion, Cross Validation, Model Complexity and Reduced Error Pruning,” in Information Technologies, IJCST, 2012.

[8] Wikipedia: http://en.wikipedia.org/wiki/Botnet

[9]RuleQuest:http://www.rulequest.com/see5comparison.html.

[10]Fortiguard:http://fortiguard.com/legacy/analysis/zeusanalysis.html

[11]Citadel: http://www.botnets.fr/index/citadel.html

[12]NaïveBayes:http://www.ic.unicamp.br/naivebayesclassifier.html