ISSN22773061

Ontological Engineering Approach Towards Botnet Detection in Network Forensics

Sukhdilpreet Kaur, Amandeep Verma
Punjabi University Regional Centre for Information Technology and Management, Mohali
Sukhdilpreet83@gmail.com
Assistant Professor, Punjabi University Regional Centre for Information Technology and Management, Mohali
vaman71@gmail.com

ABSTRACT

The abundance in the usage of Internet, in every arena of life from social to personal, commercial to domestic and other aspects of life as well, leads the rise in cybercrime at an upsetting speed. More illegal activities as a result of cyber crime, reason to tempts many network attacks and threats. Network forensics is the branch of fornesics that deals in the detection of network attacks. Botnet is one of the most common attacks, but hazardos. It is a network of hacked computers It involves the capturing, storing and then analysis of the network packets, in order to identify the source of the attack. Various methods based on this approach for botnet detection are suggested in literature but there is no generalized method to represent the basic methodology used by any of the botnet detection method. With such guidelines, the comparison among the various implementations, a roadmap for the new implementation, development of reusable implementations can be addressed. Accordingly, there is a requirement of a generic framework that can characterize the general methodology followed by any of the botnet detection methods. This paper, review various prevalent methods of botnet detection to extract commonalities among them. A global model for the detection of botnets is represented as ontology. Ontology is used as a means of knowledge representation. The botnet ontology is represented using Web Ontology Language (OWL). OWL is used because it is a language with layered architecture and high expressive power.

Indexing terms/Keywords

Network forensics, Botnet, Botnet detection methods, Ontology

Academic Discipline And Sub-Disciplines

Computer Science and Engineering

SUBJECT CLASSIFICATION

Network Forensics

TYPE (METHOD/APPROACH)

Representation

1 INTRODUCTION

Cyber crime is an alarming problem these days. In past few years many researchers have done research on network forensics to lessen the cyber crime. Network forensics is the forensic science that investigates the network traffic and analyzes it for the detection of network attacks. It also tries to find out the source of attack [1]. Botnet is one of the network attacks. It is a network of infected machines called Zombies that have their own life cycle. Botnets are controlled by a controller called botmaster. There is a need to detect the attacks and to prevent them. Detection methods detect and prevent these attacks and try to find out the source of attacks. Many methods of botnet detection are available in literature that are broadly classified into two categories Honeynet based [2] and Passive network traffic monitoring based [3]. Passive network traffic monitoring methods include Botnet Detection Through Fine Flow Classification [4], Detecting Botnets Through Log Correlation [5], Detecting Botnets with Tight Command and Control [6], Botnet Detection by Monitoring Similar Communication Patterns [7], DNS based [8], Data mining based, anomaly based and signature based [9]. All the methods have their specific framework but the generic framework is missing.

In the present study, the focus is around the implementation of general model for botnet detection method because many botnet detection methods are available in literature but there is no such generalized approach. The generic framework of botnet detection is lacking in the literature, which motivates the present study to implement a generalized model for botnet detection. It is then represented as ontology because ontology is the best means of knowledge representation. This work is indented for those researchers who want to build a new model for the botnet detection that considers the general architecture.

This paper is organized as follows: Section 2 presents the background that describes forensics, network forensics, botnets, botnet detection methods and ontology. The related work is discussed in section 3. The implementation of the generic framework of botnet detection methods is presented in section 4. Future work is stated in section 5.

2 BACKGROUND

2.1 Forensics

Forensics is the investigation technique that is used to gather evidences of some criminal activities. Forensic science have many branches and network forensics is one of them.

2.2 Network Forensics

Network forensics is a branch of forensics science and is the extension of network security. Network security simply detects and prevents the attacks but the network forensics has the capability to do investigation [10]. Network forensics is the investigation technology, which captures the network packets, record them for investigation and then analyze and correlate the recorded network data to find out the source of attacks [1].

2.3 Network attacks

With the increase usage of Internet, there is also a rapid increase in cyber crime, which includes various network attacks. Network attacks exploit the vulnerabilities of the system and gain unauthorized access to the system [11]. One of the network attacks is botnet.

2.4 Botnets

Botnet is one of the common network attacks these days. Botnet is defined as a network or group of compromised computers called zombies, which are controlled by a botmaster automatically [3]. The botmaster controls the whole botnet using Command and Control servers [9].

2.5 Botnet detection methods

Botnet detection methods detect the botnet attacks. The botnet detection methods are broadly classified into two categories Honeynet based botnet detection and passive network traffic monitoring [12]. Honeynet is made of collection of more than one honeypot and a honeywall. A honeypot is a system designed to attract the attackers so as to observe their activities and find out solutions [2]. Passive network traffic monitoring methods are further classified into IDS based detection, DNS based and data mining based detection techniques. The IDS based detection methods can be Signature based also called misuse based and Anomaly based also called network statistical behavior based. Sinature based methods are further classified into host based and network signature based.

2.6 Ontology

Ontology describes the concepts, with their properties and the relationships between the concepts, of a specific domain. It is the best means of knowledge representation. The concepts can be events or classes. The relationships are the properties among the concepts like object properties that relate objects to objects and data properties that relate object to data [13].

3 RELATED WORK

Cyber crime is a huge problem these days. In past few years many researchers have done research on Network forensics to reduce the cyber crime.

3.1 Network Forensics in literature

Ahmad Almulhem and Issa Traore [1] explored the topic of network forensics and proposed architecture of network forensics system. The proposed architecture manages to collect attack data at network and hosts. It is a capable of bypassing encryption if used by a hacker.

The challenges in deploying a network forensics infrastructure are highlighted by Ahmad Almulhem [10] in “Network Forensics: Notions and Challenges”. The various aspects of network forensics and related technologies were presented with limitations of those technologies.

3.2 Botnet and Botnet Detection in literature

J.S.Bhatia, et al [2] presented the introduction to various Internet attacks. They discuss the botnet attacks and propose an approach to detect the botnet attacks that use the IRC and HTTP protocols. The proposed approach is based on Virtual Honeynet based system. They evaluated the approach using real world network traces.

Maryam Feily, et al [3] presented a survey on botnet and botnet detection. The presented survey clarifies what is botnet and also discusses the various botnet detection techniques. Their survey divides the botnet detection techniques into four categories: DNS-based, signature based, anomaly based and mining-based. It also compares the various botnet detection techniques.

Xiaonan Zang, et al. [4] conducted an experiment to observe the discriminating capabilities of the Hierarchical and K mean clustering algorithms and exploring a RTT adjustment procedure to mix the botnet trace with the background Internet traffic. Their experiment has shown the proposed capabilities.

Yousof Al-Hammadi and Uwe Aickelin [5] proposed a new technique to detect the presence of botnets. They used an interception technique to monitor Windows Application Programming Interface (API) functions calls made by communication applications and store these calls with their arguments in log files. They proposed an algorithm to detect botnets based on monitoring abnormal activity by correlating the changes in log file sizes from different hosts [5].

Systems detect botnets by examining traffic content for IRC commands or by setting up honeynets. W. Timothy Strayer, et al. [6] proposed an approach for detecting botnets by examining the flow characteristics such as duration, bandwidth, and packet timing that looks for evidence of botnet Command and Control activity. They constructed an architecture that first eliminates traffic that is unlikely to be a part of a network of bots; the remaining traffic is classified into a group that is likely to be part of a botnet, and then correlates the likely traffic to find common communications patterns that would suggest the activity of a botnet. The main focus of this method is on reduction of data set by feeding the traffic packet traces into a series of quick reduction filters.

Hossein Rouhani Zeidanloo and Azizah Bt Abdul Manaf [7] provide taxonomy of botnets C&C channels and they also evaluate the well-known protocols that are being used. They also proposed a general detection framework that focuses on botnets based on P2P and IRC protocols. Their proposed botnet detection framework does not need any prior knowledge like signatures of the botnets.

Sandeep Yadav and A.L. Narasimha Reddy [8] explored the techniques that may utilize the failed domain queries. They present the DNS based botnet detection method.

Yousof Ali Abdulla Al-Hammadi [9] presented an approach that is host-based behavior for the detection of botnets. He monitor the function calls within a time window using various correlation algorithms. He uses an intelligent algorithm that is inspired from the immune systems.

The concepts of network attacks and network security along with cryptography are discussed in [11] by William Stallings.

Alexander V. Barsamian [12] proposed a framework to characterize the network behavior. He starts the research by collecting the network traffic from packet series and hypothesizes that they will characterize the behavior of traffic from threat data. He develops a method to measure the conformity and also detect behavioral changes and also evaluate it. He uses the Kullback-Leibler divergence method for this. He also describe various methods based on K-means approximation for detecting synchronous behavior .He analyze an application of their proposed methods and detect the hosts on the network for the presence of botnet infection.

Robert F. Erbacher, et al [14] introduced a multi-layered architecture to detect the various botnets. They use multiple techniques to detect the old as well as new botnet attacks that cannot be detected by a single technique. For the detection of well-known old botnet attacks, they use signature type techniques and for new botnets, data mining are used.

3.3 Ontology in literature

Gustavo Gonzalez Granadillo, et al [13] introduced an ontology-driven approach to address the problems of heterogeneous system devices and network to manage the security events and also select the appropriate

countermeasures. They proposed a model that considered two main aspects, the information manipulated by (Security Information and Event Management) SIEM environments and the operations that are applied to this information. A case study on botnets is presented to illustrate the utilization of proposed model. Their proposal uses shared information called ontology, between elements and classes, to ensure interoperability among the various components (like services, machines, and users) of the system and constant processes. They also provided an example of the proposed model over a botnet attack showing the functionality of main operations.

Andrew Simmonds, et al [15] introduces the network security services and review about various threats and vulnerabilities. They use the review for the construction of a framework that is used to build ontology for network security attacks.

4 IMPLEMENTATION OF GENERIC FRAMEWORK OF BOTNET DETECTION

implementation of the general model for botnet detection is proposed in this section. The proposed framework is composed of some components extracted from the review of literature. The list of the components used to implement the general model is

1. Filters
2. Classifiers
3. Correlator
4. Clusters
5. Analyzer

4.1 Implementation as Ontology

The generic framework of the model for botnet detection is implemented as ontology because is a good means of knowledge representation. The genal model of botnet detection is implemented as ontology using Protégé.

Tools used

- OWLViz plugin
- OntoGraf plugin

The ontology of proposed framework consists of the following concepts. The concepts used in the proposed ontology are classes. There are eight classes in the ontology of proposed generic framework of botnet detection.

Classes

The various classes in the hierarchy of botnet detection method are described in table 1.

Table 1: various classes and their description

Class	Description
DataSource	The various sources of the data to be analyzed like network data, file system information and system process information.
TrafficScanner	It represents the data capturing tool, which gathers the specific network information, create Log files, monitor the data and maintain a list of suspicious IP addresses
PacketFilter	It depicts the filter component of the botnet detection methods that converts the packet traces into flow summaries, detect traffic content, select the TCP based flow and filters out the handshaking process.
FlowClassifcicationEngine	It corresponds to the classifier component of the botnet detection methods that extracts and inspect the payload, does content matching and classify the flow into chat-like and non-chat like flows.
PairwiseCorrelator	It represents the classifier component, which does the pairwise examination of data and finds the correlation value.
Clustering	It denotes the cluster component and group the flows that have similar flow characteristics.
TopologicalAnalyzer	It depicts the Analyzer that identifies the controller of the botnets.
Result	It shows the details of the controller of the botnets. The details include the IP address of the bot controller alongwith the name of the bot.

Properties

The properties among the classes are described here. The ontology of the proposed generic framework consists of object properties and datatype properties.

Object properties

The various object properties that exist between the classes in the ontology of botnet detection method are described in table 2.

Table 2: various object properties and their description

Object properties	Description
isSending	It represents the relation between classes sending some data to other classes
isReceiving	It represents the relation between classes receiving some data from other classes

Datatype properties

The various datatype properties between the classes of the proposed framework are described in table 3.

Table 3: various datatype properties and their description

Datatype properties	Description
hasIPAddress	Build datatype property of integer
hasNameofBot	Build datatype property of string

Individuals

The individuals in ontology represents the objects of the class the individuals in the ontology of proposed framework are Ethreal, Sebet, Snort, RandomForest, J48DecisionTrees, SDBot,GTBot and SpyBot.

OWLViz Plugin

The generalized model is composed of eight classes a described in table 1. The classes are DataSource, TrafficScanner, PacketFilter, FlowClassificationEngine, PairwiseCorrelator, Clustering, Topological Analyzer and Result. All these classes are sub classes of the class Thing. OWLViz plugin shows the is-a relationship only twhich means that it shows the class and subclass relationships. The snapshot of the classes implemented in the proposed model of botnet detection method taken from OWLViz is shown in figure 1.

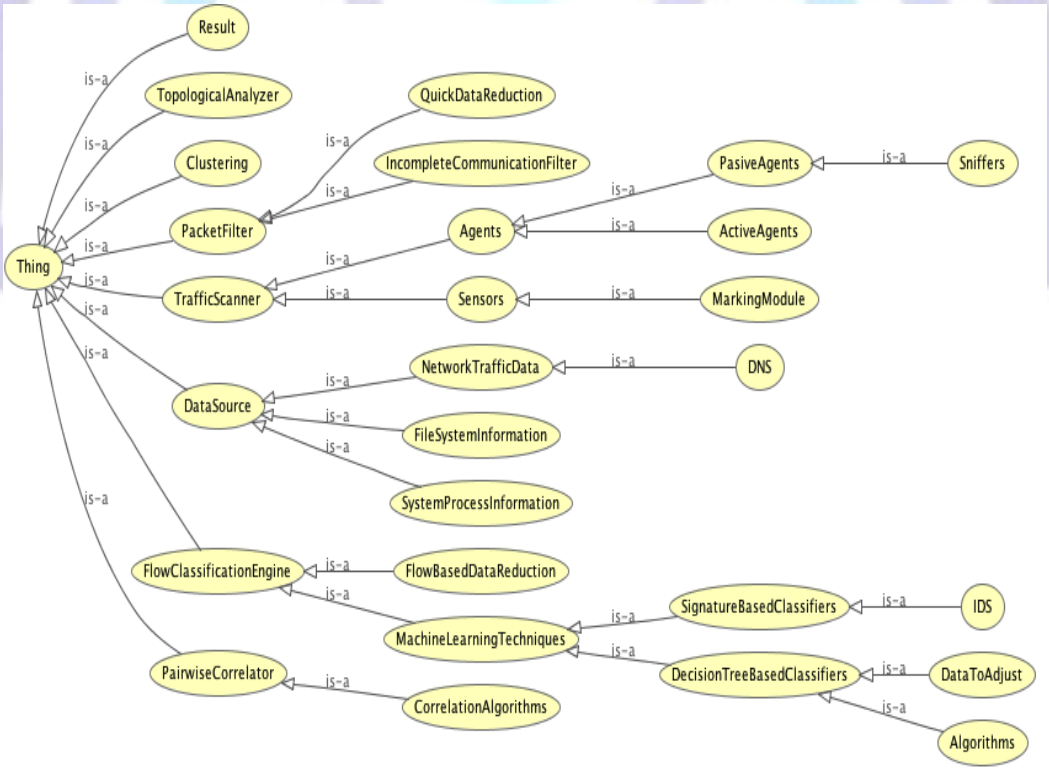

Figure 1: OWLViz plugin displaying ontology of proposed generic framework of Botnet Detection Method

The classes of the proposed model are discussed below:

DataSource: DataSource class denotes the various sources of data that are to be tested against botnet attack. It has three subclasses that are NetworkTrafficData, SystenProcessInformation and FileSystemInformation. NetworkTafficData has a subclass DNS. DataSource class sends the data to the TrafficSCanner class.

TrafficScanner: TrafficScanner class depicts the data-capturing tool that captures the traffic, monitor and examine the data in packets and mark the suspicious IP addresses. TrafficScanner is again composed of two subclasses Agents and Sensors. Agents gather specific network information. Agents can be ActiveAgents or PassiveAgents and create log files whereas the Sensors monitor the data in packets. PassiveAgents class has a subclass Sniffers that are used to capture the data and have individuals Ethreal and Sebet. The Sensors has subclass MarkingModule that marks and makes a list of the suspicious IP addresses. TrafficScanner class sends the packet traces to the PacketFilter class.

PacketFilter: PacketFilter class models the filters that are used to reduce the workload by filtering out the network traffic that are not a part of the botnets and it also convert the packet traces into flow summaries. PacketFilter class is composed of two subclasses QuickDataReduction and IncompleteCommunicationFilter. QuickDataReduction class mainly focuses on filtering out the traffic by selecting only the TCP based flows and the IncompleteCommunicationFilter focus on filtering out the handshaking process. PacketFilter class sends the remaining flows, after the process of filtration, to the FlowClassificationEngine.

FlowClassificationEngine: FlowClassificationEngine class denotes the classifiers that classify the remaining flows into groups with the help of two subclasses, which are FlowBasedDataReduction and MachineLearningTechniques. FlowBasedDataReduction class extract the payload and machineLeariningTechniques class models the content matching part of the classifiers. MachineLearningTechniques are further classified into two subclasses SignatureBasedClassifiers and DecisionTreeBasedCLassifiers. SignatureBasedClassifiers inspects the payload and has a subclass IDS, which depicts the Intrusion Detection Systems. IDS class has an individual called Snort. DecisionTreeBasedClassifiers have subclass Algorithms which models the various machine learning algorithms like RandomForest and J48DecisionTree that are individuals in ontology. The other subclass of DecisionTreeBasedClassifiers is DataToAjust class that depicts the data used to modify the learning algorithms for better results. This class classifies the flows into chat-like and non chat-like flows, and then sends the chat-like flows to the PairwiseCorrelator class.

PairwiseCorrelator: PairwiseCorrelator class describes the correlator component of the general botnet detection methods that does the pairwise correlation and finds the correlation value. It is composed of a subclass, CorrelationAlgorithm that denotes the algorithms to be used for finding the correlation value. This class sends the correlated flows to the Clustering class.

Clustering: Clustering class is used to group the network flows with similar flow characteristics that can be a part of same botnet. The clusters of similar flows are then forwarded to the TopologicalAnalyzer class.

TopologicalAnalyzer: TopologicalAnalyzer class models the analyzer of the general botnet detection method, which tries to identify the controller, means the source of botnet attack. This class after analyzing the clusters sends the details of the controller of the botnet to the Result class.

Result: Result class depicts the report generation, which is used to generate the report. Result generates a report with the details of the controller including the name and IP address of the controller. The names of bots that can be detected are the individuals of this class like SDBot, GTBot, and SpyBot.

OntoGraf Plugin

Ontograf plugin also show the object properties and datatype properties that cannot shown by the OWLViz plugin.

How to read the hierarchy

The hierarchy of the concepts that are used in generic framework of botnet detection method is presented in this section. The explanation of the symbols used in the hierarchy is described in table 4.

Table 4: symbols used in hierarchy represented by OntoGraf plugin

Symbol	Represents
Rectangle with yellow solid circle	Classes and subclasses
Rectangle with plus sign	The class is expandable
Rectangle with purple diamond sign	The individuals
Solid line	Has subclass relationship
Dotted line	Object properties

Classes

In next topic, all the classes designed in protégé are presented using the OntoGraf plugin of Protege. The classes deigned in the OntoGraf plugin is used to show those relationships which are not able to be shown by the OWLViz plugin. The OntoGraf plugin shows the subcass relationships as well as the object properties between the various classes defined in the Ontology.

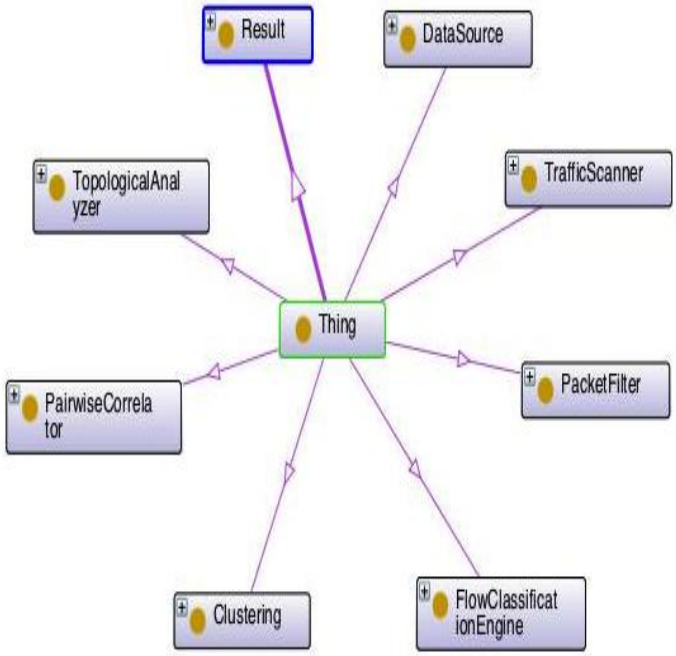

Figure 2: Classes designed using protégé

Figure 2 represents all the eight classes that are present in general model of Botnet detection Ontology. These classes depicts the various components of the generalized model of botnet detection method.

From the next figure the detailed descripton of each class will be discussed.

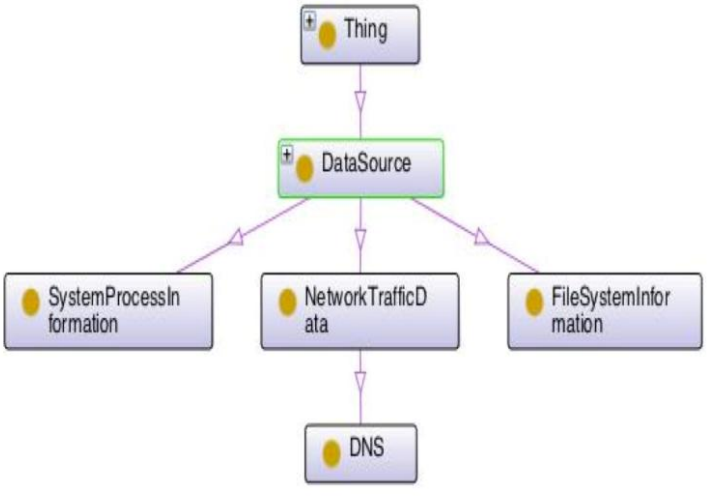

Figure 3: Subclasses of DataSource class

Figure 3 describes the subclasses of DataSource class. The subclasses of DataSource class are SystemProcessInformation, NetworkTrafficData and FileSystemInformation. NetworkTrafficData class has a subclass DNS.

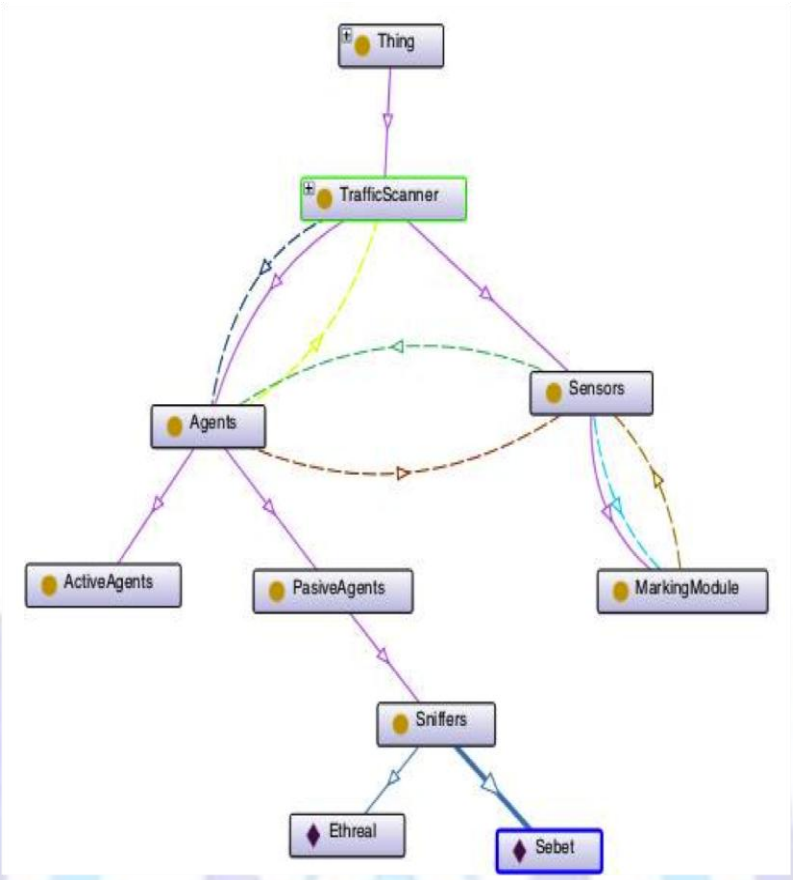

Figure 4: Subclasses of TrafficScanner class

Figure 4 shows the clas TrafficScanner that has two subclasses Agents and Sensors. The dotted line shows the object properties among the classes TrafficScanner and Agents, TrafficScanner and Sensors, Agents and Sensors, Sensors and MarkingModules. This figure also shows the objects Sebet and Ethreal of the class Sniffers, which were not shown using the OWLViz plugin of Protégé.

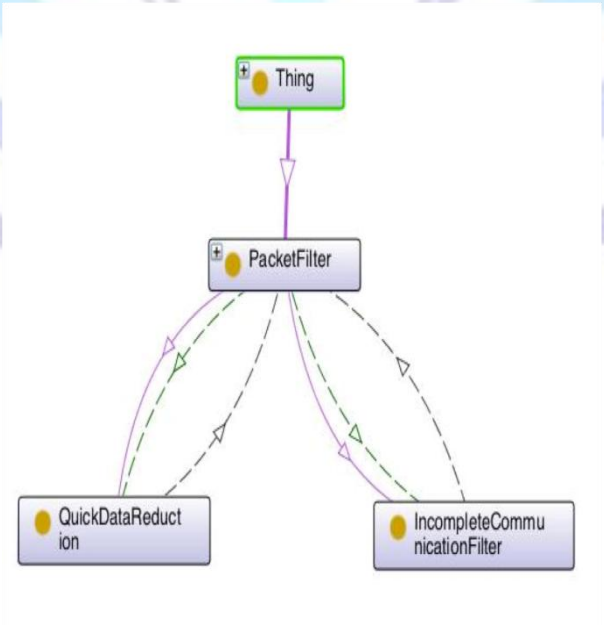

Figure 5: Subclasses of PacketFilter class

Figure 5 shows the class PacketFilter and its subclasses QuickDataReduction and IncompleteCommunicationFilter. It shows the object properites between all the classes and subclasses.

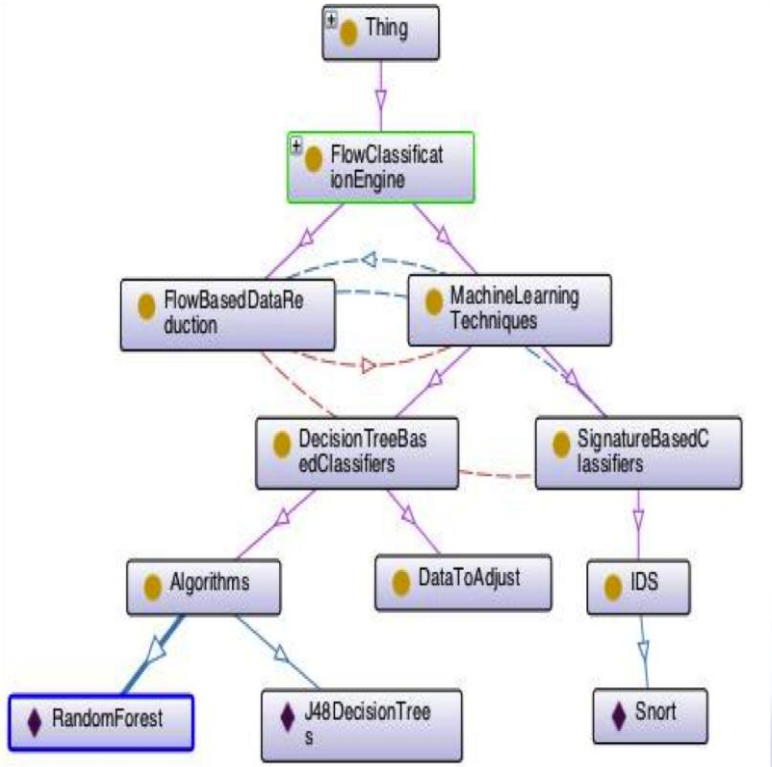

Figure 6: Subclasses of FlowClassificationEngine class

Figure 6 shows the class FlowClassificationEngine with its two subclasses FlowBasedDataReduction and MachineLearningTechniques. MachineLearningTechniques have further subclasses SignatureBasedClassifier and DecisionTreeBasedClassifier. SignatureBasedClassifiers class has a subclass IDS and DecisionTreeBasedClassifiers have subclasses Algorithms and DataToAdjust. In this figure the individuals RandomForest and J48DecisionTrees of class Algorithms and object Snort of class IDS are also shown. There are object properties between FlowBasedDataReduction and MachineLearningTechniques and also between FlowBasedDataReduction and SignatureBasedClassifier.

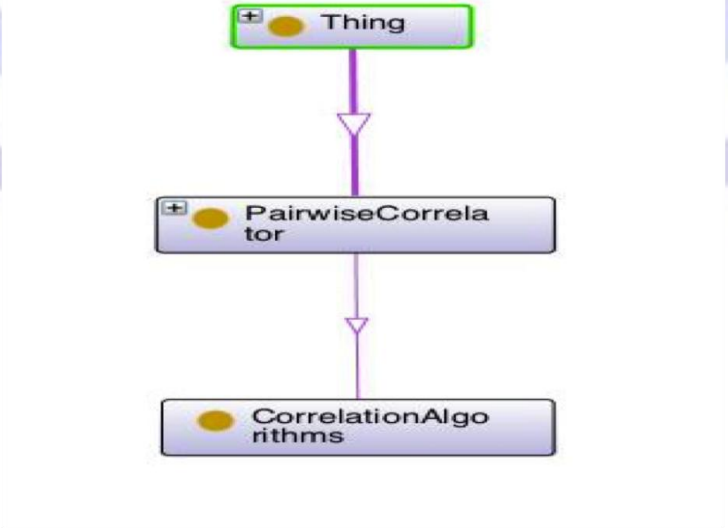

Figure 7: Subclasses of PairwiseCorrelator class

Figure 7 shows the class PairwiseCorrelator with its subclass CorrelationAlgorithm. There is no object property between them. This class finds the correlation value using the correlator algorithms.

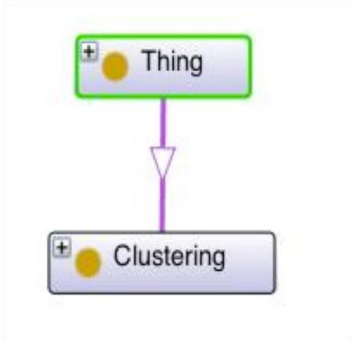

Figure 8: Clustering class

Figure 8 shows the class Clustersing as the subclass of the class Thing. This class doesnot have any subclass. This class groups the flow of traffic that have similar network characteristsics. The plus sign on the class Thing and Clustering means that these classes are expnadable on some relationships

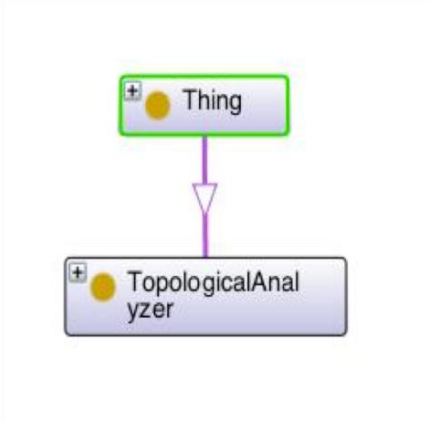

Figure 9: TopologicalAnalyzer class

Figure 9 shows the class TopologicalAnalyzer. It is the subclass of the class Thing and does not have any further subclass. The TopologicalAnalyzer class analyze the clustered traffic for the detection of Botnet attack. In this figure also, there is a plus sign on the Thing class that again denotes that this class can be expanded on subclasses or on object properties. There is a plus sign on the TopologicalAnalyzer class also which means this class is also expandable. The complete diagram with the expanded classes are presented in figure 5.11.

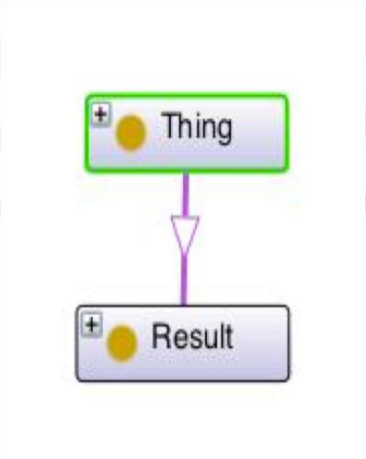

Figure 10: Result class

Figure 5.10 shows the class Result. Result class also does not have any further subclass.

All the above figures show the classes, that are the subclasses of the class thing, along with their subclasses, objects and relationships between them. But still there are some object properties that are missing to be discussed because those properties are among all the eight classes. So these properties will be shown and discussed only if the complete hierarchy of the classes using the OntoGraf plugin of protégé tool and is shown in next section.

Detailed heirarchy of the classes in the ontology of proposed framework

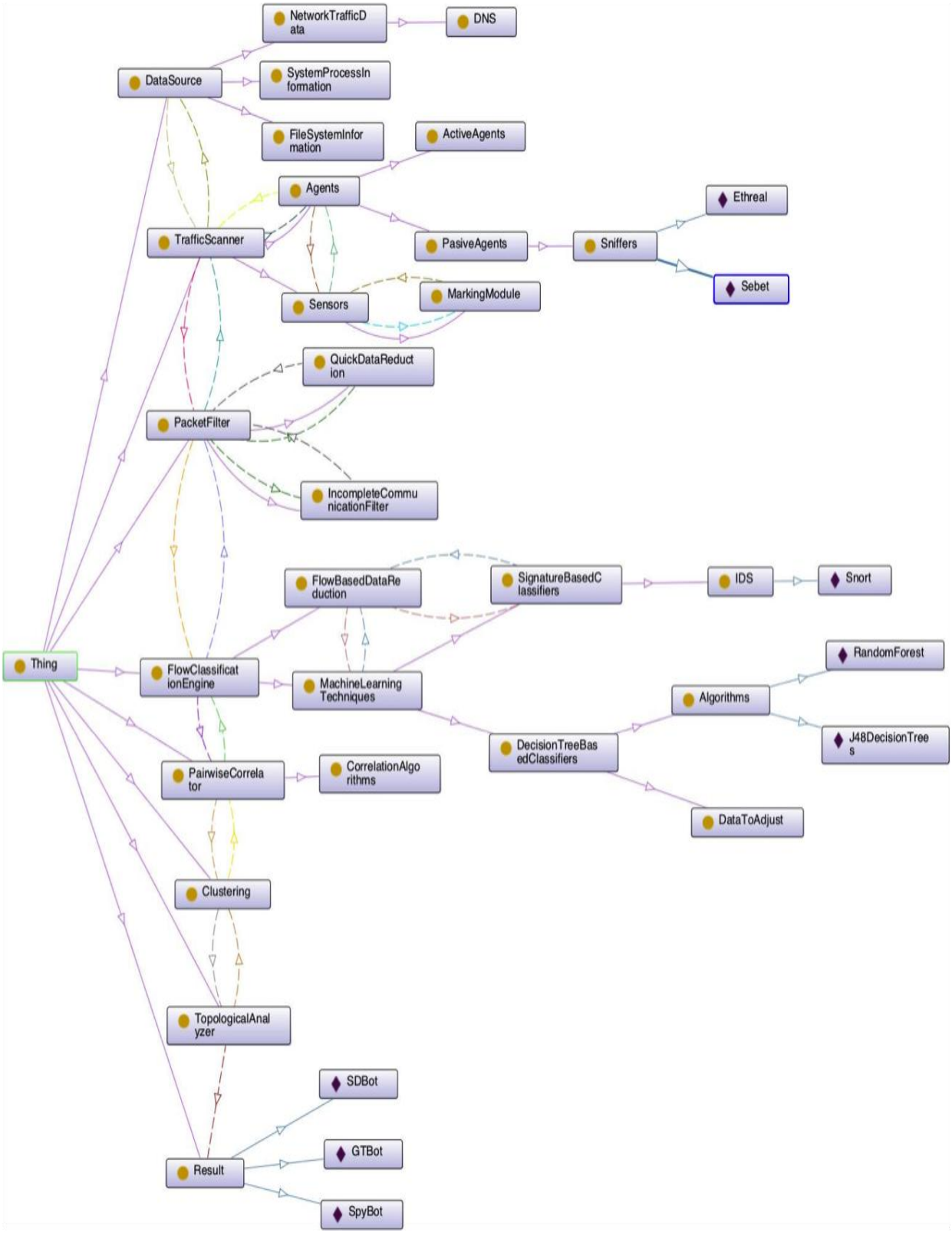

Fig 4: OntoGraf Plugin showing the generic framework of botnet detection method

FUTURE WORK

In future, the present work can expand the general Botnet detection Ontology. The Ontology can be populated and then can be used. The presented model of Botnet detection can be more generalized.

ACKNOWLEDGEMENTS

We are highly indebted to God for his blessings and love throughout my life and for not letting me down in difficult times. We are grateful to our family for their support.

REFERENCES

[1] Ahmad Almulhem and Issa Traore, "Experience with Engineering a Network Forensics System", in Proc. of the 2005 in international conference on Information Networking, pp. 62-71, 2005.

[2] J.S.Bhatia, R.K.Sehgal and Sanjeev Kumar, "Botnet Command Detection using Virtual Honeynet", in International Journal of Network Security & Its Applications, pp. 117-189, vol.3, no.5, Sep 2011.

[3] Maryam Feily, Alireza Shahrestani and Sureswaran Ramadass, "A Survey of Botnet and Botnet Detection", in Proc. Of 3rd International Conference of IEEE on Emerging Security Information, Systems and Technologies, SECURWARE '09, Athens, Glyfada, pp. 268 - 273, June 2009.

[4] Xiaonan Zang , Athichart Tangpong, George Kesidis and David J. Miller, "Botnet Detection Through Fine Flow Classification", unpublished, Report No. CSE11-001, Jan. 31, 2011.

[5] Yousof Al-Hammadi and Uwe Aickelin, "Detecting Botnets Through Log Correlation", in Proc. of the Workshop on Monitoring, Attack Detection and Mitigation (MonAM2006), Tubingen, Germany, pp. 97-100, 2006.

[6] W. Timothy Strayer, Robert Walsh, Carl Livadas and David Lapsley, "Detecting Botnets with Tight Command and Control", in Proc. of the 31st IEEE Conference on Local Computer Networks (LCN), pp. 195-202, 2006.

[7] Hossein Rouhani Zeidanloo and Azizah Bt Abdul Manaf, "Botnet Detection by Monitoring Similar Communication Patterns", International Journal of Computer Science and Information Security, pp. 36-45, vol. 7, no. 3, 2010.

[8] Sandeep Yadav and A.L. Narasimha Reddy, "Winning with DNS Failures: Strategies for Faster Botnet Detection", in Proc. of 7th International ICST Conference, SecureComm, London, UK, pp. 446-459, 2011.

[9] Yousof Ali Abdulla Al-Hammadi, "Behavioural Correlation for Malicious Bot Detection", Ph.D. dissertation, The University of Nottingham, April 2010.

[10] Ahmad Almulhem, "Network Forensics: Notions and Challenge", IEEE International Symposium on Signal Processing and Information Technology, pp. 463 - 466, 2009.

[11] William Stallings, " Cryptography and Network Security: Principles and Practices", 3rd ed. Pearson Education ©2002 ISBN:0130914290

[12] Alexander V. Barsamian, "Network Characterization For Botnet Detection Using Statistical-Behavioral Methods", M.S. thesis, Thayer School of Engineering Dartmouth College, Hanover, New Hampshire, June 2009.

[13] Gustavo Gonzalez Granadillo, Yosra Ben Mustapha, Nabil Hachem and Herve Debar, "An Ontology-based model for SIEM Environments," Springer, ICGS3/e-Democracy, vol.99, pp. 148-155, 2011.

[14] Robert F. Erbacher, Adele Cutler, Pranab Banerjee and Jim Marshall, "A Multi-Layered Approach to Botnet Detection", in Proc. of the International Conference on Security and Management, Las Vegas, Nevada, USA, pp. 301- 308, July 2008.

[15] Andrew Simmonds, Peter Sandilands, Louis van Ekert, "An Ontology For Network Security Attacks", Faculty of IT, University Of Technology Sydney.