A Probabilistic Approach to Detect and Prevent Bandwidth Depletion Attacks

Abhijit Boruah
Asst. Professor
Dept. of CSE, DUIET
Dibrugarh University, India

ABSTRACT

Capturing uncertain aspects in network security domain and their analysis by an intelligent agent is an important research domain in the current world of implementing AI in network security. When an intelligent agent is referred to, the picture that immediately comes to minds is a design that can sense the environment and take legitimate decisions by itself based upon the knowledge gathered from its environment. Hence the ability of reasoning among the agents is an important factor which governs this ability to act. There are a lots of knowledge representation schemes which are used in domain specific situations. One such situation is representing knowledge in uncertain domains. Traditional probabilistic languages lack the expressive power to handle relational domains where as classical first-order logic is sufficiently expressive, but again lacks a coherent uncertainty reasoning capability. So, an effort was made to combine both the expressiveness of first order logic as well as plausible reasoning capability of Bayesian networks in a reasoning scheme called Multi Entity Bayesian Networks (MEBN) logic. The proposal in this paper tries to detect and prevent a type of bandwidth depletion attacks (which falls in the category of DOS attacks) by filtering out the features of the network traffic relevant to these attacks and providing them as input to a MEBN model, which finally decides the fate of the traffic i.e. either it is to be allowed to enter the network or flagged as a probable threat in future and dropped.

Keywords

Keywords: Probabilistic reasoning, DDoS, Uncertain, MEBN, UnBBayes.

1. INTRODUCTION

The recent advancement of technology has enabled dealing with a wide spectrum of human needs effectively. But parallel to it another technical world of compromising information security is growing on resulting in detrimental effects. These include attack on information security, hacking and blocking of services. Denial of service (DoS) and Distributed DoS attacks are most widely prevalent network attacks of which DDoS are very hard to block as the use of numerous zombies are seen in DDoS [1]. DOS attacks mainly aims at exhausting target resources and further restricting servers to provide services. Alternatively servers could also be exhausted off their bandwidth or connection buffers with lot of bogus packets/requests. DoS attacks are currently the most exorbitant cyber crime for victim organizations.

DDoS attacks are usually fabricated by a huge numbers of hosts which acts reflectors or amplifiers of some kind which are generally known as zombies [1]. Honeynet 2005 is an example where attacker controls a large number of zombies. Attacker would usually flood the victim with numerous bogus packets until the target resource in the victim gets exhausted. In [2] the authors have discussed the modes or classification of how the various methods of attack exploit the vulnerabilities in a host. According to [2] attacks can be either network based or host based. Networks based attacks are again classified into TCP SYN Flooding, ICMP Smurf Flooding, UDP Flooding and Intermittent Flooding. Based on how attackers scans computers, how packets are forged, determining targets and aftereffects, DDoS attacks can be further classified into Scanning based, spoofing based, target based and impact based[3]. Authors in [4] describe two models of DDoS attack networks: Agent handler model and Internet Relay chat (IRC) based model. In agent handler model, the attacker operates on the client platform and takes help of softwares distributed over the internet as handlers. These handler softwares will be tried to be loaded on compromised routers or network servers carrying heavy network traffic. In the IRC based model, an internet chat communication channel is targeted by an attacker to connect the clients to the agents. Now the attacker will be made access to legitimate IRC ports to send attack commands to the agents.

If carefully observed, most of the DDoS attacks can be prevented or premature detection is possible by using some probabilistic reasoning techniques. The inspiration to use probabilistic measures in order to detect a initiation of a DDoS attack is that if, by some means, the relevant traffic features can be continuously scanned to detect some slightest deviation from general pattern, that deviation can be analyzed by a probabilistic model to predict if the traffic is a probable threat. A lot of probabilistic reasoning schemes exists which include Bayesian Networks, Dynamic Bayesian Networks, Object Oriented Bayesian Networks etc. Finding out the proper representation scheme for modeling of the DDoS attack prevention technique will be a challenge as it should be able to represent all the details and must be highly expressive. In this approach, Multi Entity Bayesian Networks (MEBN) Logic [5] is used because of its ability to represent probability distributions over analysis of arbitrary first order domain theories.

2. LITERATURE REVIEW

There have been a lot of works recently based on applications of intelligence in network security because of the growing threats and to provide intelligent security services to important data. Most of these works are again in the sub topic of designing ontology for network management which can be used by various other modules to reason upon and take appropriate measures. Authors in [6] constructed a Bayesian network based upon a graphical threat model to analyze and capture uncertain relationships. They conducted some experiments which showed that although the CPTs of a Bayesian network based security analysis tool are provided by human users, the effectiveness must be determined in a much objective way. Another work in [7] tried to improve the attack-vulnerability relationship model by including temporal aspects of the vulnerabilities and the networks. A Dynamic Bayesian Network based model was proposed which provided a framework for continuously measuring security measures in a dynamic environment. Another novel network attack graph is presented in [8] where the authors improved the likelihood weighting algorithm (an approximate Bayesian posterior inference algorithm) to resolve certain issues which restricts the use of Bayesian networks in network Intrusion detection models. The issues depicted by authors in [8] are i) Bayesian networks don't permit directed cycles which can sometimes exist in a network attack graph, 2) temporal partial ordering relations usually exist among intrusion evidence that cannot be easily modeled in a Bayesian network, and 3) inferring both the current and the future security state of a network cannot be done by a single Bayesian network. Moreover, an alternative approach to security risk assessment called Bayesian Attack Graph is introduced in [9]. The authors also proposed a method to estimate the security risks on vulnerabilities based on the Common Vulnerability Scoring System [10]. More recently, authors in [11] proposed a framework based on Multi Entity Bayesian Networks [5] to allow or deny network traffic based on the threat probability of the incoming or outgoing network packets. The core idea of the framework was based on designing a MTheory which was used to reason upon the network packet features.

3. PRE REQUISITES

3.1 Distributed DoS Attacks

A Denial of Service (DoS) attack can be characterized as an attack with the purpose of preventing legitimate users from using a victim computing system or network resource [12]. A Distributed Denial of Service (DDoS) attack is a large-scale, coordinated attack on the availability of services of a victim system or network resource, launched indirectly through many compromised computers on the Internet. The services under attack are those of the "primary victim", while the compromised systems used to launch the attack are often called the "secondary victims." The use of secondary victims in performing a DDoS attack provides the attacker with the ability to wage a much larger and more disruptive attack, while making it more difficult to track down the original attacker. Authors in [13] classified DDoS attacks into two taxonomies: Bandwidth depletion and resource depletion attacks. A bandwidth depletion attack usually works by sending huge amount of unwanted traffic to the victim network for which legitimate users are deprived of connections. Moreover, a resource depletion attack usually targets certain vulnerabilities in the resources of a host and deprives it from processing legitimate requests. Moreover, bandwidth depletion attacks can be further classified into flood attacks and amplification attacks. Figure 1 shows the taxonomy as described in [13].

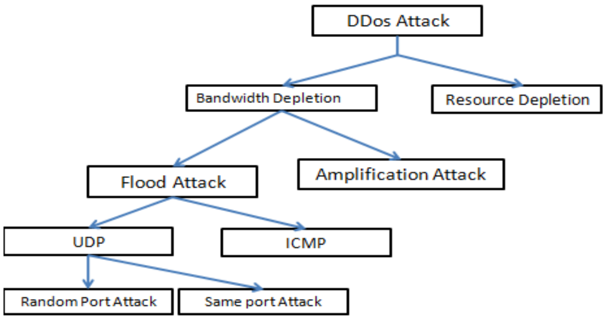


Fig. 1 Taxonomy of DDoS attacks [13]

This approach will be mainly concerned with detecting ICMP Flood attack, which a variant of bandwidth depletion attack using probabilistic reasoning.

3.2 Flood Attack

In a DDoS flood attack, the zombies flood the victim system with IP traffic. The large volume of packets sent by the zombies to the victim system slows it down, crashes the system or saturates the network bandwidth. This prevents legitimate users from accessing the victim. ICMP Flood attack is one type whose analysis would be done in this paper.

3.3 ICMP Flood Attack

Internet Control Message Protocol (ICMP) packets are designed for network management features such as locating network equipment and determining the number of hops or round-trip-time to get from the source location to the destination. For instance, ICMP_ECHO_REPLY packets ("ping") allow the user to send a request to a destination. Because ICMP can be a useful troubleshooting and diagnostic tool, it is often permitted by firewalls. Unfortunately, for the hosts behind such a firewall, bugs in the IP layers of the hosts can potentially be exploited [14].

A DDoS ICMP flood attack occurs when the zombies send large volumes of ICMP_ECHO_REPLY packets to the victim system. These packets signal the victim system to reply and the combination of traffic saturates the bandwidth of the victim's network connection.

ICMP flooding attacks are popular because of amplification techniques such as the Smurf attacks, which use a spoofed broadcast ping to generate a large number of responses that then floods a target. A typical attack scenario is shown in the figure 2 below:

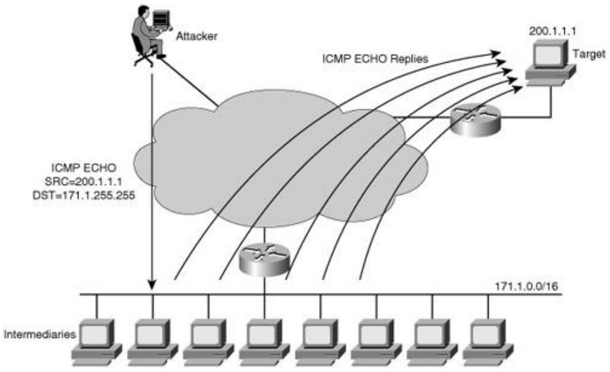


Fig 2. A Scenario of ICMP Flood attack

The attack uses ICMP echo request packets directed at IP broadcast addresses from a remote site. The intent is to cause DoS. Three parties are involved in the attacks: the attacker, the intermediary, and the victim (the intermediary can also be a victim). The intermediaries receive the spoofed ICMP echo request packets from the attacker that is directed to the IP broadcast address of the subnet. If the ICMP traffic directed to IP broadcast addresses are not filtered, the subnet will be flooded by ICMP ECHO reply messages which will overload the network [14].

3.4 Multi Entity Bayesian Networks [15]

It is a logic system that integrates First Order Logic (FOL) with Bayesian probability theory. MEBN logic provides added feature to ordinary Bayesian networks by allowing representation of complex graphical models s. Knowledge is encoded as a collection of Bayesian network fragments

(MFrags) that can be instantiated and combined to form highly complex situation-specific Bayesian networks. Semantically defined collection of MFrags specifying the conditional probabilities of events in the domain forms a MEBN theory (MTheory). MTheory implicitly represents a joint probability distribution over possibly unbounded numbers of hypotheses, and uses Bayesian learning to refine a knowledge base as observations accrue. MEBN provides a logical foundation for the emerging collection of highly expressive probability-based languages [15].

A Bayesian network is a graphical model depicting the probabilities of events occurring in future from the past events forming a directed acyclic graph. The nodes in the graph denote random variables. Local distributions specify the probabilities of random variables finally forming the joint probability distribution. Similar to Bayesian networks, MTheories also use directed graphs to specify joint probability distributions over interrelated random variables [15]. Probability information about a group of random variables is expressed via MEBN Fragments (MFrags). Random variables in MEBN logic take arguments that refer to entities in the domain of application. Boolean connectives and quantifiers in First order logic can be represented semantically by pre defined MFrags. Most importantly, any sentence in FOL can be represented by a random variable in MEBN logic. Joint probability distribution over the truth values of set of FOL sentences are expressed implicitly my MTheories.

3.5 MEBN Fragments

For An MFrag is a graphical representation of the collection of entities and their relationships in a specific domain. There are three different nodes in an MFrag: context, input and resident nodes. Author in [15] describes an MFrag as a set F = (C, I, R, G, D) consisting of:

- C → a finite set of context RVs.
- I → a finite set of input RVs.
- R → a finite set of resident RVs.
- G → a fragment graph.
- D → a set of local distributions, one for each member of R.

The context nodes are Boolean variables that represent conditions that have to be satisfied so that the probabilistic distribution of the resident nodes applies. Their possible values of context nodes are true, false and absurd. The probabilistic distribution of the child resident node is dependent on input nodes, but distributions of each of the input nodes have to be defined in other MFrags where they are resident nodes. Resident nodes have the local probabilistic distributions defined in that MFrag, including the probabilistic dependence on its parent values (that can be input or resident nodes) [15].

4. EXPERIMENTAL OBSERVATIONS AND ANALYSIS OF ICMP PING FLOOD ATTACK

Experiments were conducted over a wireless network (Wi-Fi) consisting of 8 computers to study the effects of ICMP Ping Flood Attack. The experiments were carried out in five phases.

4.1 Phase 1: Normal Traffic

In this phase the network has normal traffic that is there is no Ping Flood attack. The test bed setup is shown in Figure 3 and observation is shown in Table I. The corresponding graph of network utilization over time is shown in figure 4 for the victim computer under normal network traffic condition.

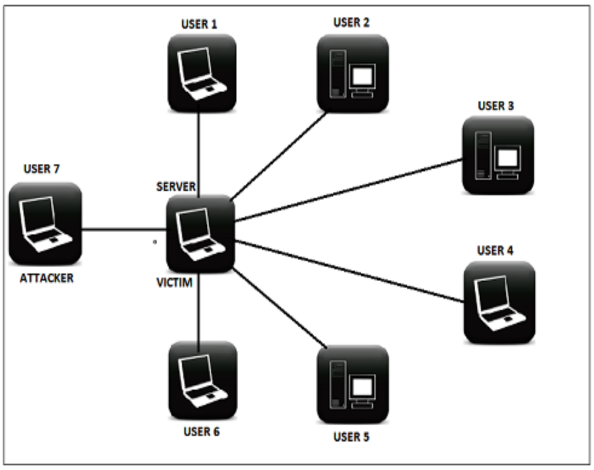
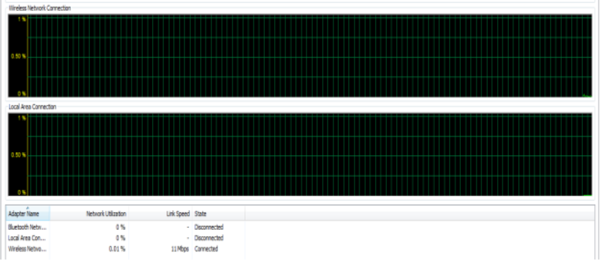


Fig 3. Network setup with no attack



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Netw...	0 %	-	Disconnected
Local Area Con...	0 %	-	Disconnected
Wireless Netwo...	0.01 %	11 Mbps	Connected

Fig 4. Network Utilization graph with no attack

4.2 Phase 2: Single Ping Flood attack with normal 32 bytes of data

In this phase the victim is been attacked with numerous Ping Request of 32 bytes of data by a single attacker. The observation is shown in I and the test bed setup is shown in figure 5. The corresponding graph of network utilization over time is shown in figure 6 for the victim computer under attack. It is observed that network utilization has increased for the victim as compared to normal network traffic.

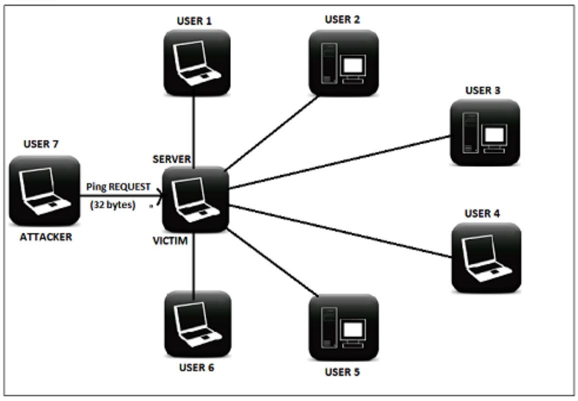
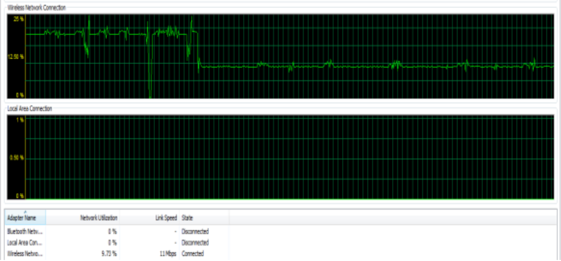


Fig 5. Network setup with Ping Flood attack with normal 32 bytes of data



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Netw...	0 %	-	Disconnected
Local Area Con...	0 %	-	Disconnected
Wireless Netwo...	9.73 %	11 Mbps	Connected

Fig 6. Network Utilization graph with single attacker (data=32bytes)

4.3 Phase 3: Multiple Ping Flood attack with normal 32 bytes of data

In this phase the victim is been attacked with numerous Ping Request of 32 bytes of data by a multiple attacker. The observation is shown in table I and the test bed setup is shown in figure 7. The corresponding graph of network utilization over time is shown in figure 8 for the victim computer under attack. Here it is observed that network utilization has increased for the victim as compared to network traffic for Single Ping Flood attack.

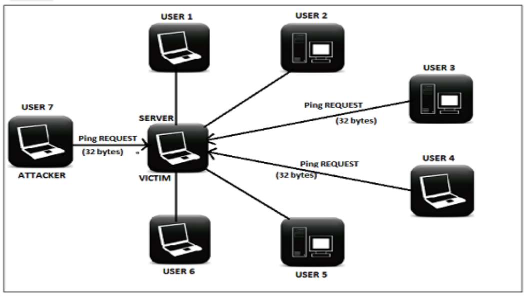
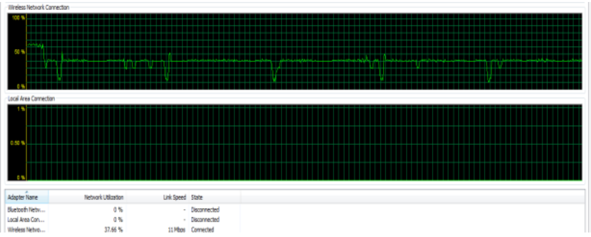


Fig 7. Network setup with Multiple Ping Flood attack with normal 32 bytes of data



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Netw...	0 %	-	Disconnected
Local Area Con...	0 %	-	Disconnected
Wireless Netwo...	37.66 %	11 Mbps	Connected

Fig 8. Network Utilization graph with multiple attacker (data=32bytes)

4.4 Phase 4: Single Ping Flood attack with 65500 bytes of data

In this phase the victim is been attacked with numerous Ping Request of 65500 bytes of data by a single attacker. The observation is shown in table I and the test bed setup is shown in figure 9. The corresponding graph of network utilization over time is shown in figure 10 for the victim computer under attack. It is observed that network utilization has increased for the victim as compared to network traffic for Single Ping Flood attack with Ping Request of 32 bytes of data.

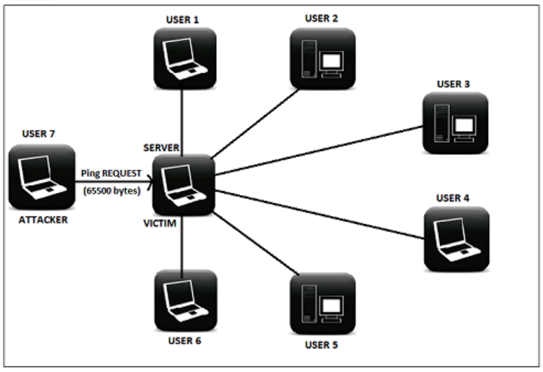
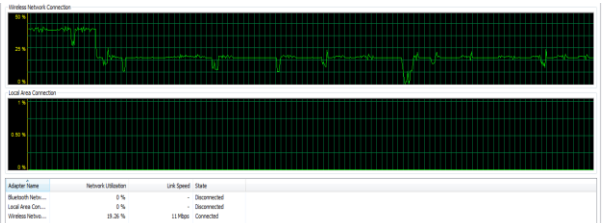


Fig 9. Network setup with Ping Flood attack with 65500 bytes of data



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Netw...	0 %	-	Disconnected
Local Area Con...	0 %	-	Disconnected
Wireless Netwo...	19.26 %	11 Mbps	Connected

Fig 10. Network Utilization graph with single attacker (data=65500bytes)

4.5 Phase 5: Multiple Ping Flood attack with 65500 bytes of data

In this phase the victim is been attacked with numerous Ping Request of 65500 bytes of data by multiple attackers. The observation is shown in table I and the test bed setup is shown in figure 11. The corresponding graph of network utilization over time is shown in figure 12 for the victim computer under attack. It is observed that network utilization has increased for the victim as compared to network traffic for any of the above phases.

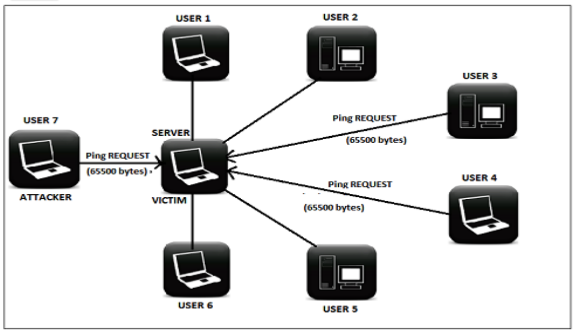
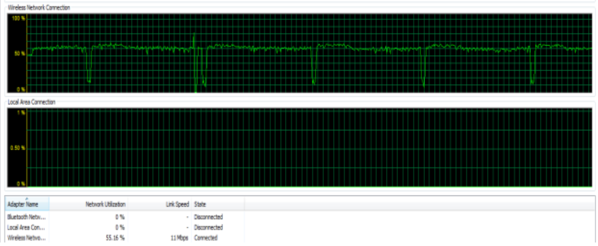


Fig 11. Network setup with Multiple Ping Flood attack with 65500 bytes of data



Adapter Name	Network Utilization	Link Speed	State
Bluetooth Netw...	0 %	-	Disconnected
Local Area Con...	0 %	-	Disconnected
Wireless Netwo...	55.16 %	11 Mbps	Connected

Figure 12: Network utilization graph with multiple attackers (data=65500bytes)

Table I shows the observations in the five experimental phases.

Table 1. Experimental Observations

Number of attacker(s)	Data size of Ping packet	Network utilization
0	**-**	**0.01%**
1	**32 bytes**	**9.73%**
3	**32 bytes**	**37.66%**
1	**65500 bytes**	**19.26%**
3	**65500 bytes**	**55.16%**

5. THE PROPOSED MEBN DECISION MODEL

Based on the model proposed by authors in [11], out threat detection model consists of three parts as show in figure 13:

- Network traffic analysis using Wireshark tool.
- User interface
- MEBN module using UnBBayes tool.

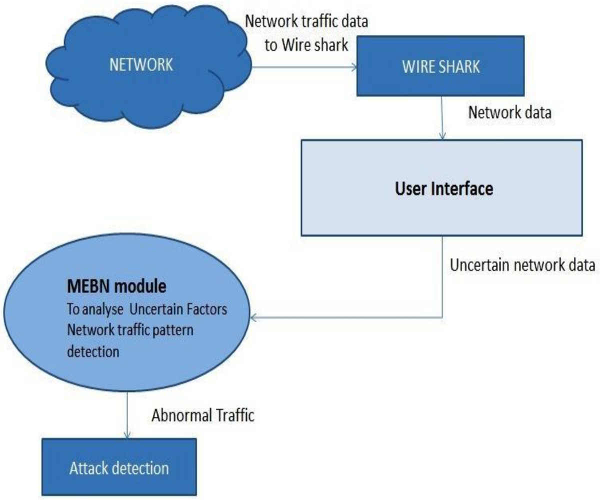
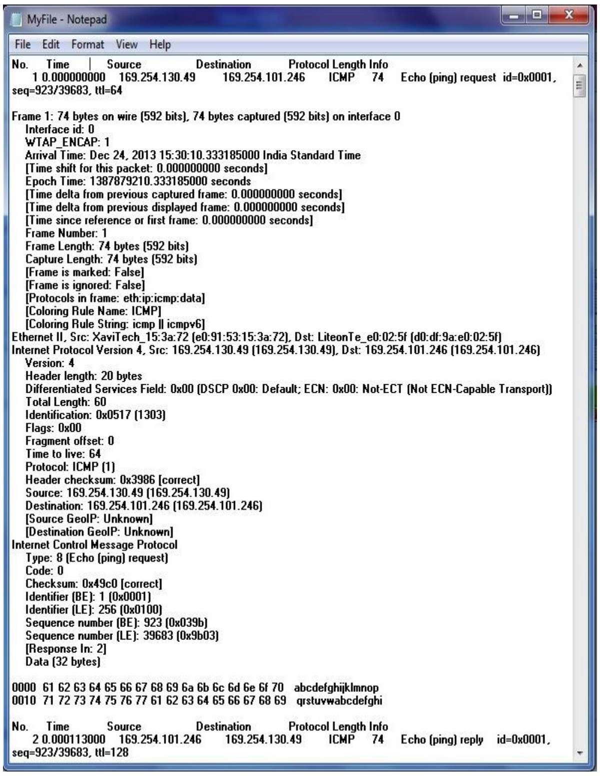


Fig: 13. Proposed Architecture

The basic idea behind the working of the proposed architecture is as follows: Network traffic entering into the subnet will be first passed through a traffic capturing module (for experimental purpose, wireshark packer analyser tool is used here). Then there is the system interface (a simple Java UI) which will extract features which have the possibility to carry information related to uncertainty from the network data and pass it to the MEBN module. After analysing, the MEBN module will determine either the traffic have the possibility of being a future attack or not.

5.1 Network traffic analysis & user interface.

The figure 14 and figure 15 shows snapshots of packet captured by Wireshark tool and filtered data from user interface respectively.



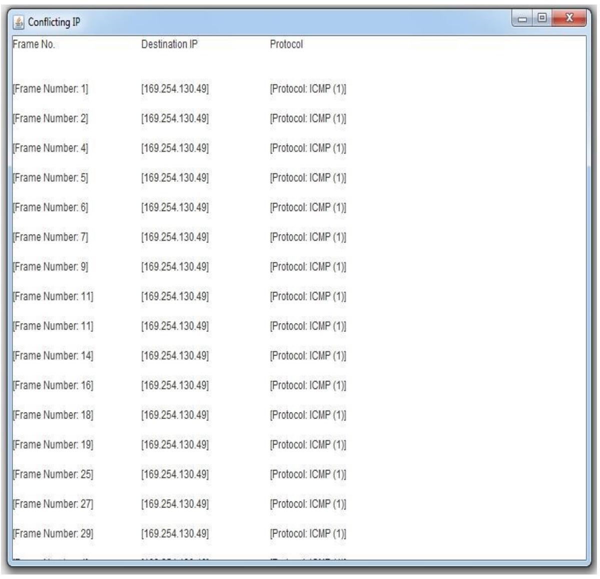
```
No.     Time    |   Source            Destination        Protocol Length Info
      1 0.000000000    169.254.130.49       169.254.101.246      ICMP     74     Echo (ping) request  id=0x0001,
seq=923/39683, ttl=64

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
    Interface id: 0
    WTAP_ENCAP: 1
    Arrival Time: Dec 24, 2013 15:30:10.333185000 India Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1387879210.333185000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: XaviTech_15:3a:72 (e0:91:53:15:3a:72), Dst: LiteonTe_e0:02:5f (d0:df:9a:e0:02:5f)
Internet Protocol Version 4, Src: 169.254.130.49 (169.254.130.49), Dst: 169.254.101.246 (169.254.101.246)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 60
    Identification: 0x0517 (1303)
    Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0x3986 [correct]
    Source: 169.254.130.49 (169.254.130.49)
    Destination: 169.254.101.246 (169.254.101.246)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x49c0 [correct]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 923 (0x039b)
    Sequence number (LE): 39683 (0x9b03)
    [Response In: 2]
    Data (32 bytes)

0000  61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70   abcdefghijklmnop
0010  71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69   qrstuvwabcdefghi

No.     Time           Source            Destination        Protocol Length Info
      2 0.000113000    169.254.101.246      169.254.130.49       ICMP     74     Echo (ping) reply    id=0x0001,
seq=923/39683, ttl=128
```

Fig: 14 Captured traffic data from Wireshark



```
Frame No.              Destination IP         Protocol


[Frame Number: 1]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 2]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 4]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 5]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 6]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 7]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 9]      [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 11]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 11]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 14]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 16]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 18]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 19]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 25]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 27]     [169.254.130.49]       [Protocol: ICMP (1)]

[Frame Number: 29]     [169.254.130.49]       [Protocol: ICMP (1)]
```

Fig: 15 : Filtered data from User Interface

5.2 MEBN modeling using UnBBayes [17]

This proposal uses UnBBayes [17] tool for modeling probabilistic ontology to detect ICMP Ping Flood attack. UnBBayes is an open source software for modeling, learning and reasoning upon probabilistic networks. It has support to Probabilistic networks (Bayesian Networks, Multi sectioned Bayesian networks, Hybrid Bayesian networks, object oriented Bayesian networks etc), FOL Probabilistic networks

(MEBN, PR-OWL), Learning Bayesian networks, sampling, Sampling and Classification Performance Evaluation [16].

The following four MFrags were designed based on the specific domain of detecting ICMP Flood attack:

- PacketOrigin MFrag
- ProtocolType Mfrag
- PacketFrequncy MFrag
- AttackIntension MFrag

Where *AttackIntension* MFrag finally provides us the decision of what is the probability of the traffic being an ICMP Ping flood attack. Following are the descriptions of the MFrags

PacketOrigin MFrag: This MFrag consists of two context random variables and one resident random variable. The context random variable *Isa(s,source_label)* and *Isa(p,packet_label)* represents the assumption that the MFrag applies to entities of type *source_label* and *packet_label*.The resident random variable *PacketOrigin(p)* defines the probability of origin of a packet of a particular protocol type from a source. The PacketOrigin MFrag is shown in figure 16

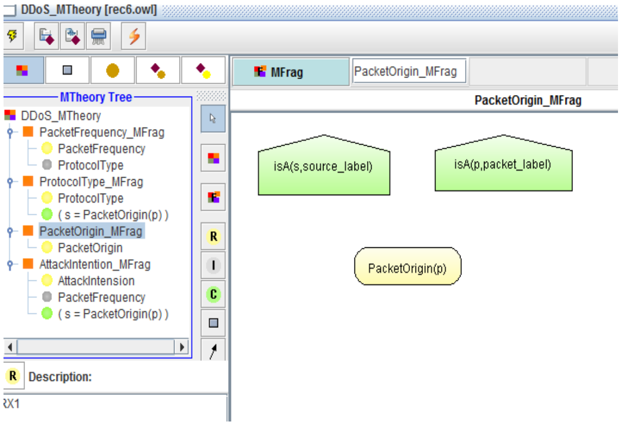


Fig 16: The PacketOrigin MFrag

ProtocolType MFrag: This MFrag consists of three context random variables and one resident random variable. The context random variable Isa(s, source_label) and Isa(p,packet_label) represents the assumption that the MFrag applies to entities of type *source_label* and *packet_label*. Given a source this MFrag represents the probability of sending packets of a particular protocol type. ProtocolType MFrag is shown in figure 17.

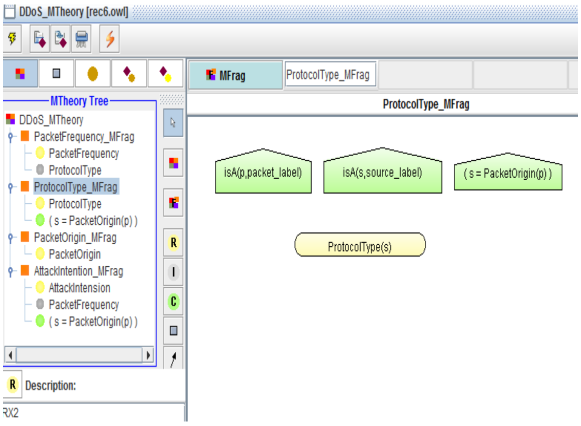
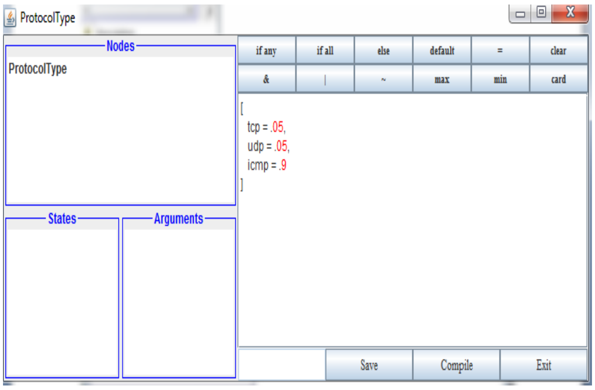


Fig: 17: The ProtocolType MFrag



```
[
 tcp = .05,
 udp = .05,
 icmp = .9
]
```

Fig: 18: Probability Distribution of ProtocolType MFrag

Figure 18 shows the probability distribution of states of node ProtocolType. These probability distribution are obtained by analyzing the experimental network traffic .ProtocolType node has three node states tcp, udp & icmp and in case of ICMP Ping Flood attack the probability of icmp protocol type is high.

PacketFrequency MFrag: This MFrag models the frequency of a source of sending packets of a particular protocol type. This MFrag as shown in figure 19 has one context variable, one input variable and one resident variable. The context random variable Isa(s,source_label) represents the assumption that the MFrag applies to entities of type *source_label*. That is, instances of this MFrag can be created by replacing the variable *s* by the identifiers of entities of type *source_label*, and as many instances can be made when the source is given. The assumption of this MFrag is that if the probability of a source of sending packets of a particular protocol type is more, then the packet frequency of the corresponding protocol type will be more.

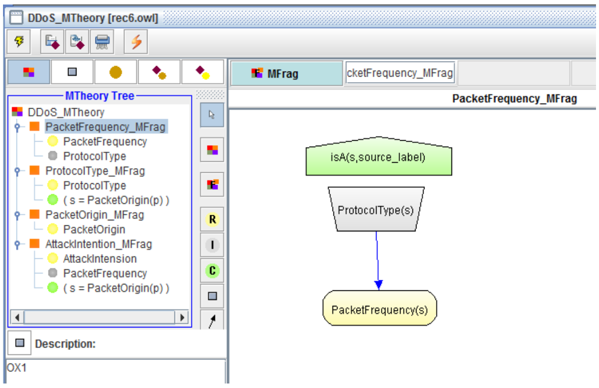
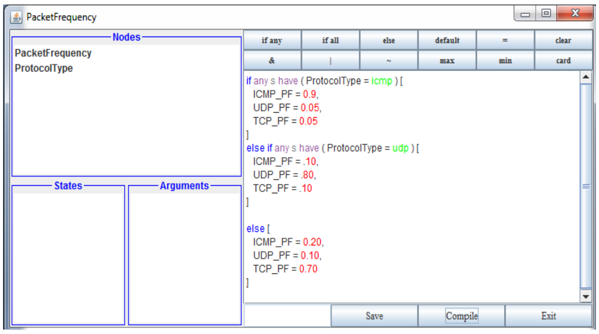


Fig: 19. PacketFrequency MFrag



```
if any s have ( ProtocolType = icmp ) [
  ICMP_PF = 0.9,
  UDP_PF = 0.05,
  TCP_PF = 0.05
]
else if any s have ( ProtocolType = udp ) [
  ICMP_PF = .10,
  UDP_PF = .80,
  TCP_PF = .10
]

else [
  ICMP_PF = 0.20,
  UDP_PF = 0.10,
  TCP_PF = 0.70
]
```

Fig: 20 Probability Distribution of PacketFrequency MFrag

Figure 20 shows probability distribution of states of node *PacketFrequency*. These probability distributions are obtained by analyzing the experimental network traffic. The conditional probability table is shown in figure 21.

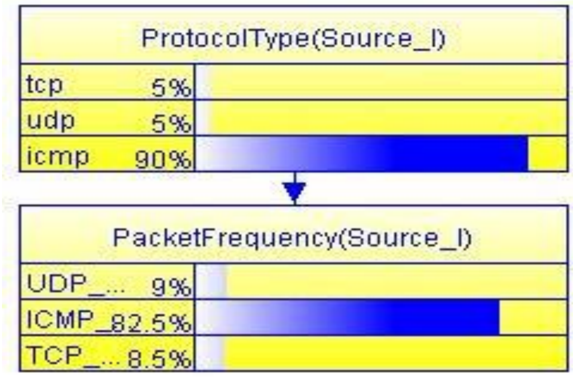


Fig: 21. CPT of PacketFrequency MFrag

AttackIntension MFrag: This MFrag as shown in figure 22, models the attack intention of a source toward the system, given frequency of packet from the source. This MFrag has three context variable, one input variable and one resident variable. The context random variable Isa(s,source_label) represents the assumption that the MFrag applies to entities of type *source_label*. That is, instances of this MFrag can be created by replacing the variable *s* by the identifiers of entities of type *source_label*, and as many instances can be made as the source is given. The assumption of this MFrag is that if the packet arrival frequency from a particular source is above a threshold for a particular protocol type then its probability of having an attack intension is more. A source with a moderate packet frequency however is expected to have very low attack intension.

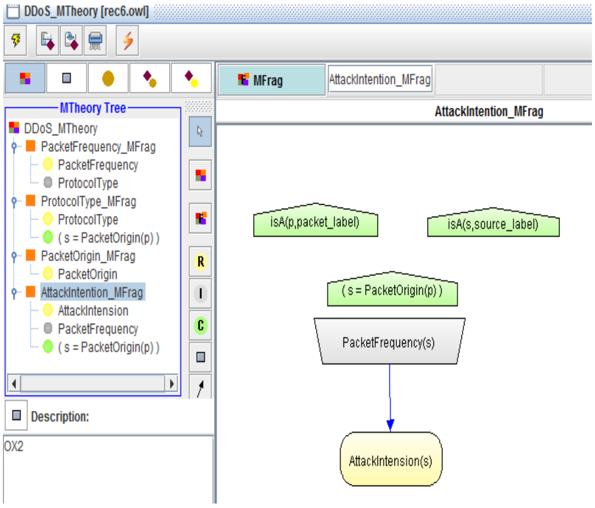
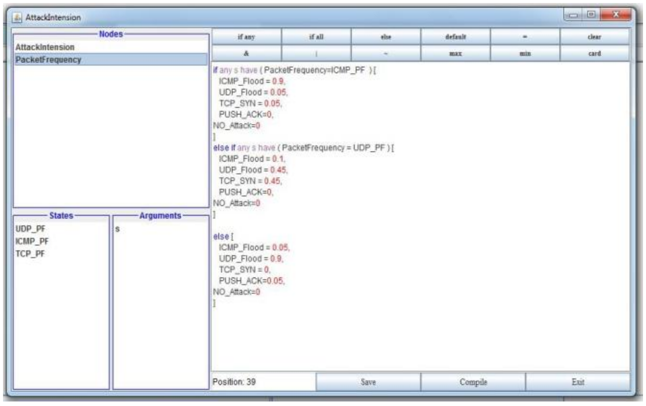


Fig: 22. AttackIntension MFrag



```
if any s have ( PacketFrequency=ICMP_PF  )[
  ICMP_Flood = 0.9,
  UDP_Flood = 0.05,
  TCP_SYN = 0.05,
  PUSH_ACK=0,
NO_Attack=0
]
else if any s have ( PacketFrequency = UDP_PF ) [
  ICMP_Flood = 0.1,
  UDP_Flood = 0.45,
  TCP_SYN = 0.45,
  PUSH_ACK=0,
NO_Attack=0
]

else [
  ICMP_Flood = 0.05,
  UDP_Flood = 0.9,
  TCP_SYN = 0,
  PUSH_ACK=0.05,
NO_Attack=0
]
```

Fig: 23. Probability Distribution of AttackIntension Mfrag

Figure 23 shows probability distribution of states of node *AttackIntention*. These probability distributions are obtained by analysing the experimental network traffic. Its Conditional Probability table is shown in figure 24.

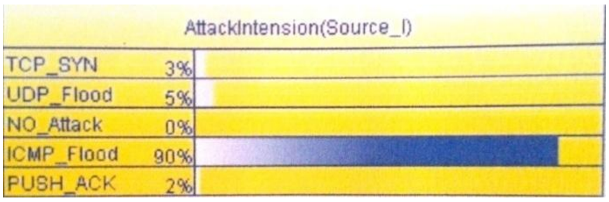


Fig: 24. CPT of AttackIntension MFrag

Hence, it is seen that the *AttackIntension* MFrag is able to decide on the percentage of probability traffic has of being an ICMP flood attack. The decision is based upon the experimental analysis of the traffic fed to the MFlags.

6. CONCLUSION & FUTURE WORKS

The aim of this work is to study and analyse detection of DoS and DDoS at the target end using Multi Entity Bayesian Networks (MEBN) logic. DDoS attacks make a networked system or service unavailable to legitimate users. These attacks are an annoyance at a minimum, or can be seriously damaging if a critical system is the primary victim. Loss of network resources causes economic loss, work delays, and loss of communication between network users. Solutions must be developed to prevent these DDoS attacks

Uncertainty is a fundamental and irreducible aspect of our knowledge about the world. Probability is the most well-understood and widely applied logic for computational scientific reasoning under uncertainty. The pattern of DDoS attack is continuously changing and evolving. Attackers create new ways to perform DDoS attack. Hence it becomes difficult to identify the attack, as it depends on uncertain factors of the network traffic. So , by including some more features into this proposed model, detection of new patterns of DDoS attacks can be made and number of novel attacks can be reduced in future by premature determination of abnormalities.

7. REFERENCES

[1] Farraposo, S., Gallon, L., & Owezarski, P. (2005). Network Security and DoS Attacks. *Feb–2005. http://www. cert. org/reports/dist_workshop. pdf.*

[2] Gu, Q., & Liu, P. (2007). Denial of service attacks. *Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications*, *3*, 454-468.

[3] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*,*34*(2), 39-53.

[4] Specht, S. M., & Lee, R. B. (2004, September). Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. In *ISCA PDCS* (pp. 543-550).

[5] Costa, P. C., Laskey, K. B., Takikawa, M., Pool, M., Fung, F., & Wright, E. J. (2005). MEBN logic: A key enabler for network centric warfare.

[6] Xie, P., Li, J. H., Ou, X., Liu, P., & Levy, R. (2010, June). Using Bayesian networks for cyber security analysis. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on* (pp. 211-220). IEEE.

[7] Frigault, M., Wang, L., Singhal, A., & Jajodia, S. (2008, October). Measuring network security using dynamic bayesian network. In *Proceedings of the 4th ACM workshop on Quality of protection* (pp. 23-30). ACM.

[8] Zhang, S., & Song, S. (2011). A novel attack graph posterior inference model based on Bayesian network. *Journal of Information Security*, *2*(01), 8.

[9] Poolsappasit, N., Dewri, R., & Ray, I. (2012). Dynamic security risk management using bayesian attack graphs. *Dependable and Secure Computing, IEEE Transactions on*, *9*(1), 61-74.

[10] Ou, X., & Singhal, A. (2011). The Common Vulnerability Scoring System (CVSS). *Quantitative Security Risk Assessment of Enterprise Networks*, 9-12.

[11] Boruah, A., & Hazarika, S. M. (2014, February). An MEBN framework as a dynamic firewall's knowledge flow architecture. In *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on* (pp. 249-254). IEEE.

[12] Karig, D., & Lee, R. (2001). Remote denial of service attacks and countermeasures. *Princeton University Department of Electrical Engineering Technical Report CE-L2001-002*.

[13] Specht, S. M., & Lee, R. B. (2004, September). Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. In *ISCA PDCS* (pp. 543-550).

[14] Cisco Ebook: Chapter 01: Introduction to Network Security Principles (Part03) (Cisco Ebook: Chapter 01: Introduction to Network Security Principles (Part03)).

[15] Laskey, K. B. (2006). MEBN: A logic for open-world probabilistic reasoning.

[16] UnBBayes - The UnBBayes Site (UnBBayes - The UnBBayes Site)

[17] Carvalho, R., Laskey, K., Santos, L., Ladeira, M., Costa, P., & Matsumoto, S. (2010). *UnBBayes: modeling uncertainty for plausible reasoning in the semantic web*. INTECH Open Access Publisher.