Towards Multisensor Data Fusion for DoS detection

Christos Siaterlis
csiat@netmode.ntua.gr

Basil Maglaris
maglaris@netmode.ntua.gr

Network Management and Optimal Design Lab
National Technical University of Athens

ABSTRACT

In our present work we introduce the use of data fusion in the field of DoS anomaly detection. We present Dempster-Shafer's Theory of Evidence (D-S) as the mathematical foundation for the development of a novel DoS detection engine. Based on a data fusion paradigm, we combine multiple evidence generated from simple heuristics to feed our D-S inference engine and attempt to detect flooding attacks.
Our approach has as its main advantages the modeling power of Theory of Evidence in expressing beliefs in some hypotheses, the ability to add the notions of uncertainty and ignorance in the system and the quantitative measurement of the belief and plausibility in our detection results.
We evaluate our detection engine prototype through a set of experiments, that were conducted with real network traffic and with the use of common DDoS tools. We conclude that data fusion is a promising approach that could increase the DoS detection rate and decrease the false alarm rate.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General-Security and Protection

General Terms

Measurement, Security

Keywords

Denial of Service, Anomaly detection, Data fusion

1. INTRODUCTION

We are considering the Internet increasingly often as a standard utility, like electricity or telephone access. Reliability of its offered services becomes then critical and even a short downtime can cost hundreds of dollars. Distributed Denial of Service attacks [1] are the main threat for such cut-offs, especially because they are planned and executed by wicked individuals. In the 2000-2003 period we had several examples of DDoS attacks (against one of the largest anti-spam black-list company [5], against the "Al-Jazeera" news network [26], against the root name servers [19]) that highlight their use in electronic warfare. Judging from the latest trend to use worms as DDoS attack agents[7], the future looks bleak.

We argue that having reliable DoS detection mechanisms is a necessary step for all DoS mitigation approaches. Today high false alarm rates and successful detection only when damage is already done (near the vicinity of the victim where the available bandwidth has already been consumed in the upstream path) are the main problems that hinder the automatic deployment and the effectiveness of countermeasures like firewall filtering, rate limiting [11] or route blackholes[9]. Trying to move the countermeasures from the victim near the sources of the attack with techniques like "IP traceback" [28] or "IP Pushback" [20] is also a doubtful approach in a diverse networked world, like the Internet, because automated large scale cooperation is needed. Other solutions like spoofing prevention techniques (like Ingress [16] and RPF filtering [10]) are useful but can only discourage a potential attacker by making traceback easier.

An interesting potential is to detect and filter DoS attacks on high bandwidth, overprovisioned backbone links of ISP's. Detection in this scenario is challenging as congestion is no longer a detection criterion.

Today network engineers use custom detection methods via traffic monitoring [2] and most of the existing detection techniques are weak as they utilize thresholds on single metrics. To address this problem we utilize a data fusion algorithm. Based on the "Theory of Evidence" we combine the output of several sensors that use simple heuristics in order to detect attempted DoS attacks on a high bandwidth link that can sustain the packet floods without severe congestion. Our sensors are autonomous but are collaborating by sharing their beliefs about the network's state, ie whether it's under an attack or not. We view the network as a system with stochastic behavior without assuming any underlying functional model. The attempt to infer the unknown system state is based on knowledge reported by sensors, that may have acquired their evidence based on totally different criteria. Possible sources of information are signature-based IDS, custom DDoS detection programs, SNMP-based net-

[1] We will refer with the term DoS attack to packet flooding attacks and not to logical DoS attacks that exploit certain OS or application vulnerabilities regardless if the attackers are truly distributed in the network topology.

SAC '04, March 14-17, 2004, Nicosia, Cyprus

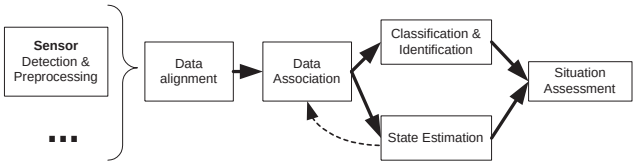

Figure 1: Typical data fusion system architecture

work monitoring systems, active measurements or accounting systems like Cisco's Netflow [8]. Our detection principle which combines the reports of various network sensors differs from the existing detection techniques that are focused on a single metric.

This paper is structured as follows: we begin with a brief introduction to data fusion systems and a short justification for our data fusion algorithm choice (sec.2). A more detailed presentation of "Theory of Evidence" and its mathematical foundations follows, particularly in contrast to "Bayesian Inference", a traditional data-fusion approach (sec.3). In section 4, we analyze the architecture of our detection engine prototype. Based on a set of experiments in an academic-network ISP, we carried out an evaluation of our detection engine implementation (sec.5). Before we conclude, we summarize the main advantages and disadvantages of our approach.

2. INTRODUCTION TO DATA FUSION

Generally, data fusion is a process performed on multi-source data towards detection, association, correlation, estimation and combination of several data streams into one with a higher level of abstraction and greater meaningfulness. In simpler words it's the process of collecting information from multiple and possibly heterogeneous sources and combining it in order to get a more descriptive, intuitive and meaningful result. Some of the most common data fusion systems are military systems for threat assessment and weather forecast systems.

The relevance of 'data fusion' with the main problem that current state of the art intrusion detection systems face has been mentioned in [4]. Our innovation consists in the use of a typical data fusion algorithm to develop a DoS detection engine that can combine the knowledge gathered by independent sensors and many different detection approaches in a powerful way and under a clear mathematical framework.

The main processing stages in most data fusion systems (Fig.1) are: the "data collection phase" where various sensors monitor, detect and report the environmental state, the "data alignment & association phase" where the collected data is aligned in time, space or measurement units, the "state estimation phase" where based on a model of the system behavior and the knowledge acquired by the sensors a data fusion algorithm estimates the state of the system, the "attribute classification & identification phase" where we identify the different targets and events that are being monitored and finally the "situation assessment phase" where the highest level of information fusion is performed. For more details we refer the reader to references [21] and [18].

We reviewed many data fusion algorithms [2] based on their applicability in the area of DoS attacks detection and we concluded that a promising method is Dempster's-Shafer's "Theory of Evidence". One of the main reasons that leaded us towards the D-S approach is that we don't have a good model for the normal network state, so we excluded physical methods, like the Kalman filter that requires the knowledge of the state transition matrix. We avoided also methods that need training data, like neural networks, because representative data of a normal state (in terms of traffic trends or other attributes) is hard to obtain and time consuming to construct. Additionally there is a clear need to utilize information from multiple heterogeneous sources with different sensitivity, reliability and false alarm rates; for example anomaly detection heuristics that go beyond signature based methods. Expert knowledge, acquired by network administrators, should be feasible to be incorporated into the system but our detection decisions should not totally rely on it or require the development of complex sets of rules that describe network behavior (like an Expert System). We could argue that these algorithms could be very useful in terms of the individual sensors detection functionality but we prefer a more flexible modeling approach for data fusion.

[2]following a taxonomy that was proposed by Hall [18]

3. MATHEMATICAL FOUNDATIONS

Our brief presentation of the "Theory of Evidence" will serve only as an introduction to the basic mathematical notations and concepts and will attempt to set the background for our application: the development of a DDoS detection engine. To complement our presentation and highlight the descriptive and modeling power of the theory, we will first present the Bayesian method for estimation that is a traditional modeling approach and has been used for DoS detection in [25]. To ease the reader we will note here, that in our application field, the observed system is the network and the measurements of the deployed sensors serve as evidence.

3.1 Bayesian inference

Let the possible states of a system be $\theta_1, ..., \theta_N \in \Theta$ and that these states are mutually exclusive and complete (exhaustive), which means that the system is certainly in one and only one of these states. The Probability $P(\theta_1)$ is an expression of the belief that the system is in state θ_1 in absence of any other knowledge. Once we obtain more knowledge in form of an evidence E then the appropriate expression to associate with the proposition θ_1 is the conditional probability $P(\theta_1|E)$ also called *"posterior probability"*. Based on the definition of conditional probabilities we have $P(\theta_1|E) = \frac{P(\theta_1,E)}{P(E)}$. Bayes theorem dictates:

$$P(\theta_1|E) = \frac{P(E|\theta_1)P(\theta_1)}{\sum_{i=1}^{N} P(E|\theta_i)P(\theta_i)} \quad (1)$$

If we have multiple evidence $E_1, ..., E_M$ and assume statistical independence between them, then we can combine them to infer the state of the observed system, similarly. We have to note that this method needs the knowledge of the "a priori" probability distribution of the states:$P(\theta_1), ..., P(\theta_N)$. In addition it does not provide any information about the quality of the result of our calculations, in terms of our trust in our evidence or the existence of conflicting evidence.

3.2 Dempster-Shafer's Theory of Evidence

Dempster-Shafer's Theory of Evidence can be considered an extension of Bayesian inference. There are many differ-

ent ways to interpret the basic mathematical formulations of the theory that was introduced by Shafer in 1976 [29]. It can be viewed either from a probabilistic or an axiomatic point of view and all these approaches are concisely surveyed in [22]. Besides the different theoretical approaches and interpretations, all of them boil down to the same mathematical formulas. Theory of Evidence has been analyzed in the fields of statistical inference, diagnostics, risk analysis and decision analysis. Our approach and notations resemble mostly the field of "Diagnostics" [30].

Let us have a set of possible states of a system $\theta_1, \theta_2, ..., \theta_N \in \Theta$,which are mutually exclusive and complete (exhaustive). The set Θ is often called *the frame of discernment*. We will call hypotheses H_i subsets of Θ, in other words elements of the powerset 2^Θ.

Our goal is to infer the true system state without having an explicit model of the system, just based on some observations $E_1, ..., E_M$. These evidence can be considered as hints (with some uncertainty) towards some system state. Based on one evidence E_j we assign a probability that it supports a certain hypothesis H_j. A *basic probability assignment (bpa)* is a mass function m which assigns beliefs in a hypothesis or as Shafer stated "the measure of belief that is committed exactly to H" [29].

$$m : 2^\Theta \rightarrow [0, 1] \tag{2}$$

This membership function m has to satisfy the following conditions:

$$m(\emptyset) = 0 \; and \; m(H) \geq 0, \forall H \subseteq \Theta \; and \; \sum_{H \subseteq \Theta} m(H) = 1 \tag{3}$$

At this point we have to underline the flexibility and advantages of this theory in contrast to the Bayesian approach, where we can only assign probabilities on single elements of Θ and not on elements of the powerset of the possible states. This theory gives us the opportunity to model uncertainty and the fact that some observations can distinguish between some system states, while they might not be able to provide any hints about others. For example, we might know that an evidence points to hypothesis $H = \{\theta_1, \theta_2\}$ with a high probability but on the same time it provides no information (complete ignorance) whether the system is in θ_1 or θ_2.

Furthermore it is crucial that the "Theory of Evidence" calculates the probability that the evidence supports a hypothesis rather than calculating the probability of the hypothesis itself (like the traditional probabilistic approach).

We define a *belief function* Bel, describing the belief in a hypothesis H, as:

$$Bel(H) = \sum_{B \subseteq H} m(B) \tag{4}$$

This definition says intuitively that a portion of belief committed to a hypothesis B must also be committed to any other hypothesis that it implies, ie to any $H \supseteq B$.H A Belief function has the following properties:

$$Bel(\emptyset) = 0 \; and \; Bel(\Theta) = 1$$

The *Plausibility* of H is defined as

$$Pl(H) = \sum_{B \cap H \neq \emptyset} m(B) \tag{5}$$

and can be correlated to the doubt in the hypothesis H:

$$Pl(H) = 1 - Doubt(H) = 1 - Bel(H^c) \tag{6}$$

where H^c is the complement of H. Intuitively, this relation means that the less doubt we have in a hypothesis H the more plausible it is. Generally we can characterize Bel(H) as a quantitative measure of all our supportive evidence and Pl(H) as a measure of how compatible our evidence is with H in terms of doubt. The true belief in H lies in the interval [Bel(H),Pl(H)]. Our degree of ignorance is represented by the difference Bel(H)-Pl(H).

The second important element of Dempster-Shafer theory is that it provides a *rule to combine* independent evidence E_1, E_2 into a single more informative hint $m_{12} = m_1 \oplus m_2$.

$$m_{12}(H) = \frac{\sum_{B \cap C = H} m_1(B) m_2(C)}{\sum_{B \cap C \neq \emptyset} m_1(B) m_2(C)} \tag{7}$$

Based on this formula we can combine our observations to infer the system state based on the values of belief and plausibility functions. In the same way we can incorporate new evidence and update our beliefs as we acquire new knowledge through observations. Theory of Evidence makes the distinction between uncertainty and ignorance, so it's a very useful way to reason with uncertainty based on incomplete and possibly contradictory information extracted from a stochastic environment. It does not need "a priori" knowledge or probability distributions on the possible system states like the Bayesian approach and as such it is mostly useful when we don't have a model of our system. In comparison with other inference processes, like first order logic which assumes complete and consistent knowledge and exhibits monotonicity [3] or probability theory which requires knowledge in terms of probability distributions and exhibits non-monotonicity [4] , Theory of Evidence has a definite advantage in a vague and unknown environment. The main disadvantage of Dempster-Shafer's theory is the assumption that the evidence are statistically independent from each other, since sources of information are often linked with some sort of dependence.

The "Theory of Evidence" from a computational point of view is in worst case exponential, because Dempster's rule of combination (Eq. (7)) requires finding all pairs of sets a,b such that $a \cap b = c$ which is $o(2^{|\Theta|-|c|} \times 2^{|\Theta|-|c|})$. Thus it may be hard to compute in the general case, although some efficient algorithms for fast computation exist. Nevertheless for many practical applications with few focal elements, a brute force approach is still feasible.

4. THE D-S DETECTION ENGINE

Based on the "Theory of Evidence", we have implemented a prototype for our novel DDoS detection engine that might aid network engineers to monitor their network more efficiently and with small set up cost. Our system fuses the knowledge collected from the reports of various sensors, in order to infer the state of the monitored network. Our sensors try to leverage on what network operators empirically know as signs of flooding attacks. These signs or evidence in the D-S notation, mostly stem from network monitoring systems and are very simple in nature because network engineers have feasibility as their primary concern. But these signs are not always accurate or definite indications. They are mere hints and there is a clear need to integrate them

[3] if a fact is believed it cannot be refuted, so our knowledge always increases

[4] $P(A|E_1E_2)$ not determined by $P(A|E_1)$

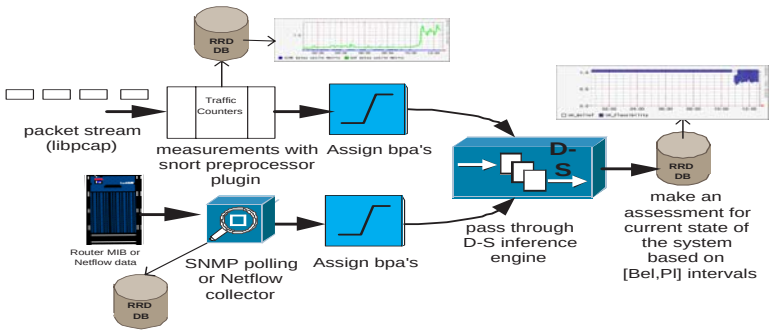

Figure 2: System architecture

into a single higher level indication. Our system's architecture is depicted in Fig.2.

As in any data fusion system, our DDoS detection system's performance depends on the selection of its sensors. The most obvious source of knowledge acquisition is passive network monitoring. Other sensor types might make active measurements like RTT or packet loss estimation [17]. Additional information can also be gathered from the Management Information Bases (MIB) that routers maintain or Netflow accounting systems that provide flow level information about network traffic. Generally speaking, some of the constraints in the selection of our sensors are that they have to be simple, efficient and easy to set up. The sensors that we have implemented so far can be classified in two different types:

- A preprocessor plugin for Snort (the popular open source IDS [6]) that produces traffic statistics based on captured packet data (libpcap format).The statistics kept were chosen to be simple so that it's feasible to run at high wire-speeds with minimum packet drops. We collect data of the incoming and outgoing TCP,TCP SYN, TCP FIN, UDP,ICMP packet rates and their corresponding share of the link utilization.
- A SNMP data collector and analyzer that stores the acquired data in round robin databases (using the RRD-tool [27]). Some examples of variables that we measure are number of active flows, flow learn failures (based on Netflow [8]) and queue drop counters.

All sensors have their own 'intelligence' based on expert knowledge. In other words they have build-in functionality, so that after the right configuration and fine-tuning they are able to express beliefs about the network state by translating their measurements to 'basic probability assignments' (bpa's).

In our simplified implementation we define the following network states that are based on a flooding attack categorization of the DDoS tools that are currently in use[23]: Θ ={NORMAL, SYN-flood, UDP-flood,ICMP-flood}. SYN attacks are targeted towards specific services mainly aiming at OS resource consumption and the rest of the attacks base their success on the sheer volume of the generated traffic, thus bandwidth consumption. We have to note here, that this set of network states (Frame of Discernment in 'Theory of Evidence' terminology) must be the same throughout the system, from the sensors to the fusion node.

In the early stages of our prototype implementation we

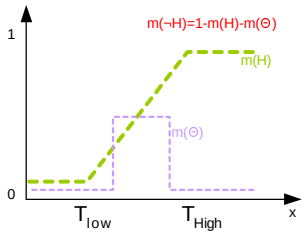

Figure 3: A generic guideline or 'rule of thumb' to define bpa's (bpa:basic probability assignment)

have implemented a sensor that is able to detect UDP flooding attacks as a change in the transferred UDP bit rate. Let us illustrate the sensor's functionality (transforming measurements to bpa's): if in one sampling interval our sensor measures a 'suspiciously high' value of the following metric $x = \frac{incoming\ UDP\ bytes/sec}{outgoing\ UDP\ bytes/sec}$ [5]then it states its increased belief in the $H_1 = \{UDP\}$ attack state. To be more specific, a sensor defines a m-value for 3 possible sets:

- It assigns a value that expresses its support for a set of states H that the sensor can recognize or is sensitive to: $m(H)$
- It assigns a value to the set $\neg H$, to express the refuting evidence of the hypothesis H: $m(\neg H)$.
- It assigns a value to the set Θ to express the ignorance of the sensor and the possibility that it might be erroneous (proportional to the false alarm rate):$m(\Theta)$.

It follows from the equation (3) that $m(H) + m(\neg H) + m(\Theta) = 1$. A guideline to help us define the individual m-values based on a measured value x is shown in Fig.3. The intuition behind this 'rule of thumb' is that although going over and under certain thresholds leads us towards a quite certain decision, in the interval between these low and high thresholds our beliefs should be treated with an increased uncertainty. We would like to discuss here the effect of changing the T_{high}, T_{low} thresholds on a sensor's performance. Assuming a typical sensor, moving the curve up and to the left yields a more sensitive sensor increasing possibly the false positive alarm rate. On the other hand a down or

[5]The DWard project [24] uses a similar metric and more specifically UDP $\frac{incoming\ packets/sec}{outgoing\ packets/sec}$. Our ratio is more effective because UDP attacks as bandwidth consumption attacks use larger packets.

right movement makes the sensor less sensitive, increasing the false-negatives. By condensing the curve along the x-axis we move towards binary detection, whereas expanding it yields a sensor with greater uncertainty but more sensitivity.

Another kind of sensor, monitors the number of active flows[6] seen by a router. Although this metric cannot give us an insight of the exact attack type it might still be a good indication of a spoofed attack [2]. In this case the modeling power of "Theory of Evidence" is apparent and the sensor states its high belief in the hypothesis H_2 ={SYN-flood, UDP-flood,ICMP-flood}.

Altogether our sensors periodically measure, calculate the corresponding bpa's and transfer the collected knowledge to the fusion node based on a communication protocol that has as its main information a *bpa* or an *m*-function definition in the form:

$< Timestamp >$:

$m < sensorid > (< hypothesisset >) =< value >$

This information can be easily expressed in XML and carried over an extension of the standard IDS communication protocol IDMEF [12].

The periodic sensor's reports update the current knowledge-base (belief pool) of the fusion node that runs with the sampling period of the fastest sensor (time alignment). The fusion node that implements Dempster's rule of combination was programmed in C and calculates the belief intervals for each member of the Frame of Discernment. All the calculated data including both the sensors statistics and the belief and plausibility values are recorded in Round Robin Databases. This way we can keep data with a configurable granularity and precision over different time scales without growing needs in storage. In the same time we use a averaging function to filter out short fluctuations. The belief intervals that are visually presented with automatically generated graphs, quantify the validity of our results. The interpretation of the results is left to the human operator and from all the possible hypothesis sets, special attention must be paid to the sets: $NORMAL$,$\neg NORMAL$, and the individual attack states.

5. PROTOTYPE TESTING

To test and evaluate our D-S detection engine prototype we have performed a series of experiments on an academic research network. As we argued in the introduction, DoS detection on an over-provisioned high-bandwidth link where traffic is aggregated but stays in low utilization levels is of great interest. In practice, a single hosted network with a fast upstream link had to be monitored for the sake of our prototype evaluation. The Gigabit Ethernet link between an ISP and a university was a good candidate. Some interesting information is that the monitored link belongs to the largest customer of the ISP and keeps a sustained rate of 200Mbps with peaks higher than 300Mbps. Additionally it contained a rich network traffic mix carrying both standard network services like web traffic, but also peer-to-peer application traffic, online games, as well as streaming audio traffic (Fig.5). This fact is significant because many of the heuristics could be sensitive to specific anomalous-looking but otherwise frequent network traffic. In other words, some detection algorithms might work in simulation or lab-testbed experiments, but their high false alarm rate in the face of real traffic would render them useless.

We conducted more than 40 experiments over several days during business hours and with background traffic generated from the more than 4000 hosts of the university campus. In our experiment scenario the victim is located inside the campus with a 10Mbps link whereas the attacker was outside the campus coming directly from the ISP. The attacker was connected to a Fast Ethernet interface (100Mbps) to simulate the aggregation of traffic from several attacking hosts and was running well known DDoS tools like Stacheldraht [14] and TFN2K [3]. We performed a series of flooding attacks with spoofed IP's [7] like SYN-floods, UDP and ICMP attacks. The network topology of our experiment setup is shown in Fig.4.

The information sources that our sensors were build upon were our Snort plugin monitoring the Gigabit Ethernet link and the MIB entries from the backbone router that were polled by our SNMP collector. The router had Netflow enabled, so that we could have access to flow level statistics.

We will present here some representative experiment results

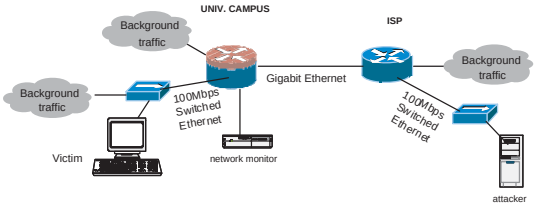

Figure 4: The topology of the experiment setup

protocol	packets/sec	Mbps
tcp	37184.8(92.55%)	204.47(94.32%)
ftp	1155.3 (2.87%)	8.19 (3.78%)
smtp	168.6 (0.42%)	0.94 (0.43%)
http/s	4011.6 (9.98%)	21.19 (9.78%)
nntp	362.9 (0.90%)	1.76 (0.81%)
p2p	7536,8 (18,75%)	41.53 (19,16%)
other	23911.9 (59.54%)	130.60(60.25%)
udp	2854.1 (7.10%)	12.23 (5.64%)
dns	180.9 (0.45%)	0.19 (0.09%)
realaud	1312.1 (3.27%)	9.93 (4.58%)
other	1,361.10 (3.38%)	2.11 (0.97%)
icmp	111.2 (0.28%)	0.075 (0.03%)
Avg: 216.80Mbps	Stddev:6.53M	Peak: 237.13Mbps

Figure 5: Partial analysis of typical traffic mix on the monitored link during a 30 min time interval

that highlight that even if one sensor fails to detect an outgoing attack, combined knowledge gathered from other sensors indicate the increased belief on an attack state clearly. In this experiment a UDP attack flooded the victim with a 34Mbps packet stream. As we see in Fig.7 the active flows metric failed to identify the attack because the spoofing mechanism was choosing source and destination ports from a limited range. Nevertheless the $\frac{incoming\ UDP\ bytes/sec}{outgoing\ UDP\ bytes/sec}$ ratio successfully identified an anomaly (Fig. 6). These sensor measurements were then translated and expressed as bpa's that are shown in figures 8 and 9. The fusion node that combined the reported beliefs generated the higher level network state representation that we can see in Fig. 10. With

[6]A flow is defined as a unique set of the following 5 characteristics <protocol, src IP, src port, dest IP, dest port>

[7]Spoofing was performed by selecting source IP's from the attacker's real subnet in order to bypass any e-gress or RPF filtering.

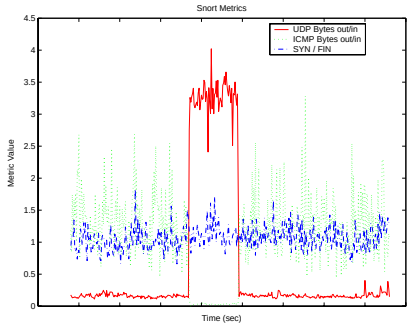

Figure 6: Output from Snort-plugin that shows that in/out UDP and ICMP bytes/sec are good heuristics for UDP attacks.

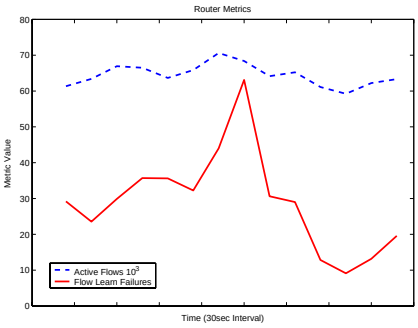

Figure 7: Output from SNMP collector that shows that the 'Flow learn failure' heuristics partially detected the attack but the 'Number of active flows' metric failed.

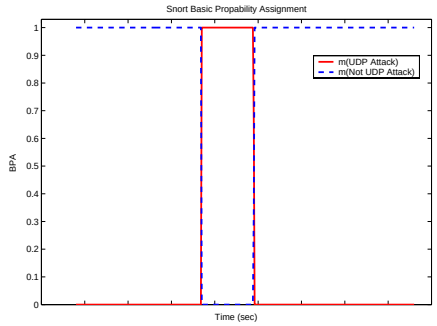

Figure 8: The basic probability assignment that corresponds to figure 6

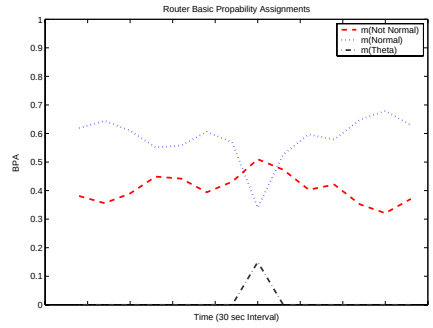

Figure 9: The basic probability assignment that corresponds to figure 7

this picture as an input, the human operator could easily identify a potential UDP attack and start an in depth exploration that will begin with an evaluation of the individual sensor reports. Our experience with the implemented detection engine showed that it's feasible to adjust the thresholds of our sensors (after a couple of experiments and with the visual aid of the automatically generated graphs) in a way that they will detect attempted flooding attacks successfully without being too sensitive. In our setup, measuring the false positive or false negative alarms is very challenging because we monitored real network traffic. The experiments that each one lasted a few minutes couldn't provide us with reliable indications. We had to develop a methodology to estimate our system's false alarm rate. Anyway, for this small time span the probability of capturing an attack that wasn't initiated by us was minute. Beyond these facts, the need of estimating our system's false alarm rate was still apparent.

Towards this goal, many open questions came into surface. How useful is the notion of the false alarm rate? How can we measure it in an unknown environment which is not controlled by us? And at the very end, do we aim to detect even unsuccessful DDoS attack attempts as anomalous events? Although we don't give final answers to these questions we present here some of our observations.

False alarm rate has been encountered in the literature as a mere percentage, a value representing the ratio of false indications in the total alarms. Another aspect of a system's false alarm rate that has been neglected but has great practical importance is its absolute value in human time scale. This means that it has to be a relatively small number in a period of several hours or days to be of any use and get analyzed and evaluated by a human expert. This way, if the total alarm rate (true positives + false positives) is a small number, we could capture some context information that may help us to do a post-mortem analysis. By analyzing the collected data, for example packet captures, we could manually guess whether the triggered alarm was false or not, estimating this way our system's false alarm rate in real, operational conditions.

Using this methodology we ran our detection engine enhanced by an automatic alarm-triggered monitoring process for a period of 28 days. We kept the sensors configuration fixed and at the fusion node we considered as an alarm a belief of an attack state with a value greater than 0.7. The results were promising as all our attacks were detected successfully and we had an average of 2.3 distinct alarms per day (Fig.11). The detailed analysis of the captured traces showed clearly that in all cases an anomaly had triggered our detection heuristics. But besides obvious packet floods generated by common file-sharing applications, the rest of the alarms were highly suspicious and although we could not argue that they were definitely DoS attacks, they could be filtered without major concerns. These results are promising for the development of automatic reaction mechanisms.

Another important concern about the effectiveness of our implementation is the performance of our passive monitoring sensor that was based on Snort. The main disadvantage of the packet capturing approach for traffic monitoring

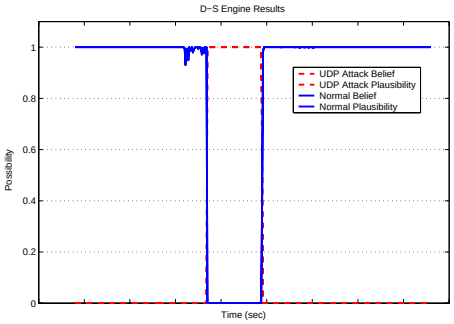

Figure 10: The output of the fusion node that combined the beliefs of figures 9 and 8

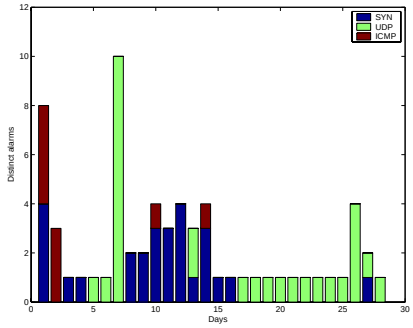

Figure 11: Distinct alarms during 28 days of operation

is the poor performance at high bit rates. Special attention had to be paid on the performance of our Snort plugin which imposed constraints on the choice of our heuristics. All the heuristics that were implemented by our Snort plugin proved to be realistic as we had negligible packet drops at 250Mbps line speed without special hardware or NIC drivers. It's important to note that in [13] we have clear indications that such simple setups can scale up to Gigabit speeds. Additionally our analysis showed that we can keep data in our RR databases in many timescales ranging from 1 second to several minutes without any change in our packet drop percent($< 0,1\%$). Finally our data fusion engine didn't encounter any performance penalty even without the use of any optimization techniques for the implementation of Dempster's rule of Combination due to the small dimension of Θ.

6. DISCUSSION

The evaluation of the proposed algorithms and architecture can be the topic of a lengthy discussion. Implementing and incorporating these ideas into the security infrastructure of an operational network may be a task of significant difficulty, but at the same time it may offer several advantages like those summarized below.

- Our modeling approach allows each sensor to contribute information at its own level of detail by expressing beliefs on a set of possible system states even without being able to asses its single elements. This fact enables us to use sensors like CPU utilization of routers that are not specific to a certain attack type.
- We don't need to assume anything about the probability of the system being on a certain state, i.e. how often attacks occur. We just express beliefs that a monitored event supports a state. Such statistics are site dependant and would be very hard to obtain.
- We can use the representation of ignorance to incorporate the false alarm rate or the predicted accuracy of a sensor to lower the false alarm rate of the fused reports. Based on the idea that a sensor performs historically in similar ways in similar situations we can use the historically-estimated correctness rate as the reference of how much should we trust the current estimation of a sensor.[31]
- We can utilize multiple data sources to increase our confidence in our estimation.
- Based on the generic representation of knowledge in terms of basic probability assignments we can incorporate knowledge from sensors based on different detection algorithms or even traditional network monitoring infrastructure like service disruption alerts. This way we can leverage on promising detection algorithms that have already been proposed. We use DS not only because classical Bayesian method is hard to apply due to the lack of a proved probability distribution model or insufficient mathematical analysis but exactly in order to be able to incorporate heterogeneous, expert knowledge into the system.
- We can activate detection algorithms on demand, to refine our beliefs. This applies especially to sensors with high processing costs that can provide new evidence on request, when it's really needed.
- The mathematical notation of membership function definition can be used to found the basis of a communication protocol for IDS collaboration as it allows fusion of data from diverse sensors. As collaborating IDS's are being explored, IETF has developed the "Intrusion Detection Exchange Protocol" [15] to help the exchange of information between different IDS's and encourage the analysis of information from different sources. The main open issue is the way that all the exchanged data is going to be combined and Theory of Evidence can provide the underlying data fusion framework.

One common drawback of knowledge-based systems is that they can be as good as the sources from which they acquire their knowledge. Utilizing expert knowledge of network administrators might not be enough. One of the strengths of our approach is that we are able to incorporate any successful detection algorithm that has been proposed in the literature by simply adding a layer of abstraction in terms of basic bpa's. The heuristics we used in our simplified prototype will be replaced with more sophisticated ones in the future. The most important fact is, that reports from other scientific fields like traffic incident detection [1] or equipment condition monitoring [30] indicate that using Dempster-Shafer to combine results of different detection algorithms increases the detection rate and simultaneously lowers the false alarm rate. One other point that can be considered as a weakness of the proposed modeling framework is its inability to detect multiple simultaneous attacks, as we assume a mutually exclusive set of system states. On the other hand we can expand the set Θ to resolve this problem.

7. CONCLUSION

We propose the use of Dempster-Shafer's Theory of Evidence as the underlying data fusion model for creating a DDoS detection engine. The modeling strength of the mathematical notation as well as the ability to take into account knowledge gathered from totally heterogeneous information sources are only some of the advantages. To demonstrate our idea we have developed a prototype that consists of a Snort preprocessor-plug in and a SNMP data collector that provide the necessary input that through heuristics feed the D-S inference engine and we evaluated our implementation trough a set of experiments in an academic research network. This simple but powerful data fusion paradigm can potentially include many of the proposed DDoS detection algorithms with their own strengths and weaknesses and could provide new solutions to the DDoS mitigation problem. It's a fact, that only if we have reliable detection mechanisms, automatic response could be a viable solution.

8. ACKNOWLEDGEMENTS

We would like to thank the personnel of the Network Operation Center of NTUA that provided us access to the necessary network infrastructure and supported our research efforts with guidelines, comments and helpful ideas.

9. REFERENCES

[1] B. Ahn and S. Byun and D.B. Choi. Traffic incident detection using evidential reasoning based data fusion. In *Proceeding of the 6th World Congress on Intelligent Transport Systems*, Toronto, Canada, 1999.

[2] P. Barford and D. Plonka. Characteristics of network traffic flow anomalies. In *Proceedings of the First ACM SIGCOMM Internet Measurement Workshop*, pages 69–74, New York, Nov. 1–2 2001. ACM Press.

[3] J. Barlow and W. Thrower. TFN2K - an analysis, March 2000. `http://packetstormsecurity.com/distributed/TFN2k_Analysis-1.3.txt`.

[4] T. Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99–105, Apr. 2000.

[5] Broadbandreports.com. Osirusoft MIA? Spammers cripple popular blacklist. `http://www.broadbandreports.com/shownews/31856`.

[6] Snort: The open source network intrusion detection system. `http://www.snort.org`.

[7] CERT/CC advisory w32/blaster worm, Aug. 2003. `http://www.cert.org/advisories/CA-2003-20.html`.

[8] CISCO. Netflow. `http://www.cisco.com/go/netflow`.

[9] CISCO. Remote triggered black hole filtering. `ftp://ftp-eng.cisco.com/cons/isp/security/`.

[10] CISCO. Unicast reverse path forwarding enhancements for the ISP-ISP edge. `ftp://ftp-eng.cisco.com/cons/isp/security/URPF-ISP.pdf`.

[11] CISCO. Using CAR during DoS attacks. `http://www.cisco.com/warp/public/63/car_rate_limit_icmp.html`.

[12] D. A. Curry and H. Debar. Intrusion detection message exchange format data model and extensible markup language (XML) document type definition. Internet Draft, Nov. 2002. Work-in-progress.

[13] L. Deri. Passively monitoring networks at gigabit speeds using commodity hardware and open source software. In *Passive and Active Measurement Workshop 2003*. NLANR/MNA, April 2003.

[14] D. Dittrich. The Stacheldraht distributed denial of service attack tool. `http://staff.washington.edu/dittrich/misc/stacheldraht.analysis`.

[15] B. S. Feinstein, G. A. Matthews, and J. C. C. White. The intrusion detection exchange protocol (IDXP). Internet Draft, Oct. 2002. Work-in-progress.

[16] Ferguson and Senie. RFC2827 network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing, May 2000.

[17] A. Habib, M. Hefeeda, and B. Bhargava. Detecting service violations and DoS attacks. In *NDSS Conference Proceedings*. Internet Society, 2003.

[18] D. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Norwood, Massachusetts, 1992.

[19] ISC/UMD/Cogent Events of 21-Oct-2002 `http://d.root-servers.org/october21.txt`.

[20] J. Ioannidis and S. M. Bellovin. Implementing pushback: Router-based defense against DDoS attacks. In *Proceedings of NDSS Symposium*, San Diego, California, February 2002. The Internet Society.

[21] J.Llinas and E. Waltz. *Multisensor Data Fusion*. Artech House, Norwood, Massachusetts, 1990.

[22] J. Kohlas and P. Monney. Theory of evidence - a survey of its mathematical foundations, applications and computational analysis. *ZOR- Mathematical Methods of Operations Research*, 39:35–68, 1994.

[23] J. Mirkovic, J. Martin, and P. Reiher. A taxonomy of DDoS attacks and DDoS defense mechanisms. Technical report 020018.Computer Science Dept., University of California, Los Angeles.

[24] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proceedings of ICNP 2002*, pages 312–321, Paris, France, November 2002.

[25] S. Mohiuddin, S. Hershkop, R. Bhan, and S. J. Stolfo. Defending against a large scale DoS attack. *Proceedings of the 3rd Annual IEEE Information Assurance Workshop*, June 2002.

[26] ITworld. Al-Jazeera hobbled by DDoS attack. `http://www.itworld.com/Sec/3834/030327aljazeera`.

[27] T. Oetiker. About RRDtool. `http://people.ee.ethz.ch/~oetiker/webtools/rrdtool`.

[28] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*.

[29] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[30] K. Tomsovic and B. Baer. Fuzzy information approaches to equipment condition monitoring and diagnosis. *Electric Power Applications of Fuzzy Systems, IEEE Press*, pages 59–84, 1998.

[31] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang. Sensor fusion using Dempster-Shafer theory. In *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, Anchorage, AK, USA, 2002.