Fuzzy Online Risk Assessment for Distributed Intrusion Prediction and Prevention Systems

Kjetil Haslum, Ajith Abraham and Svein Knapskog
Center for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology
O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
haslum@q2s.ntnu.no, ajith.abraham@q2s.ntnu.no, knapskog@q2s.ntnu.no

Abstract

A Distributed Intrusion Prediction and Prevention Systems (DIPPS) not only detects and prevents possible intrusions but also possesses the capability to predict possible intrusions in a distributed network. Based on the DIPS sensors, instead of merely preventing the attackers or blocking traffic, we propose a fuzzy logic based online risk assessment scheme. The key idea of DIPPS is to protect the network(s) linked to assets, which are considered to be very risky. To implement DIPPS we used a Distributed Intrusion Detection System (DIDS) with extended real time traffic surveillance and online risk assessment. To model and predict the next step of an attacker, we used a Hidden Markov Model (HMM) that captures the interaction between the attacker and the network. The interaction between various DIDS and integration of their output are achieved through a HMM. The novelty of this paper is the detailed development of Fuzzy Logic Controllers to estimate the various risk(s) that are dependent on several other variables based on the inputs from HMM modules and the DIDS agents. To develop the fuzzy risk expert system, if-then fuzzy rules were formulated based on interviews with security experts and network administrators. Preliminary results indicate that such a system is very practical for protecting assets which are prone to attacks or misuse, i.e. highly at risk.

1. Modelling of DIPPS

1.1. Introduction

Intrusion Prevention Systems (IPS) are proactive defense mechanisms designed to detect malicious packets embedded in normal network traffic and stop intrusions dead, blocking the offending traffic automatically before it does any damage rather than simply raising an alert as, or after, the malicious payload has been delivered. There are a number of challenges for the implementation of an IPS device that does not come across when deploying passive-mode Intrusion Detection System (IDS) products. These challenges all stem from the fact that the IPS device is designed to work in-line, presenting a potential choke point and single point of failure. Some of these problems could be eliminated in a distributed intrusion prevention system, where there is not a single point of control and the problems are tackled as close to its source of origin as possible. The main task of the IPS is to discard all suspect packets immediately and block the offending traffic flow as soon as possible. The suspicious traffic may be re-routed to honeynets or honeypots for further forensic analysis etc. An IPS should have a maximum up time since it has the potential to close a vital network path and thus, once again, causing a Self Denial of Service (SDoS) condition. IPS should be computationally light and also achieve high packet processing rates since it is essential that its impact on overall network performance is minimal. The IPS should minimize false positives since this can lead to a SDoS. The IPS should be able to decide exactly which malicious traffic is blocked and also provides a mechanism for alerts and forensic analysis capabilities. Rest of the article is organized as follows. Section 2 introduces DIPPS followed by HMM in Section 3. Fuzzy modeling is illustrated in Section 4 and experiment results are given in Section 5 followed by some Conclusions.

2. Distributed Intrusion Prediction and Prevention Systems (DIPPS)

DIPS are simply a superset of the conventional IPS implemented in a distributed environment. We consider IPS as an integrated IDS with many additional functions as listed in Section 1.1. Due to the distributed nature of IPS, the implementation poses several challenges. The IDSs are embedded inside software mobile agents and placed in the net-

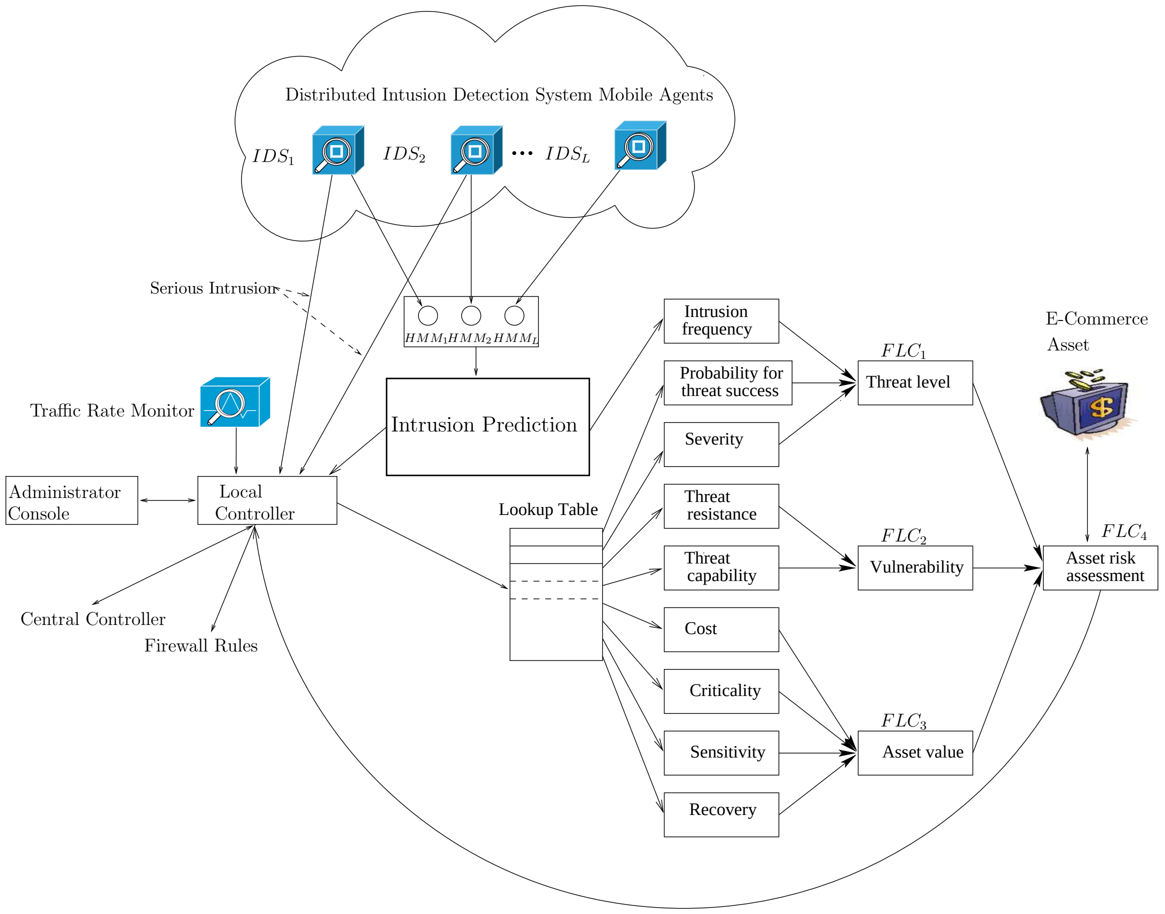


Figure 1. Architecture of a DIPPS element.

work to be monitored. An individual IDS may be configured to detect a single attack, or it may detect several types of attacks. Figure 1 illustrates the basic architecture of a DIPPS element, which is controlled by a local controller. In a large network, each DIPPS element communicates/coordinates with other DIPPS local controllers and/or a central controller. The HMM model processes the attack data information from the various mobile agent IDS sensors. IDS deployed are capable of detecting simple problems as well as serious denial of service type of attacks. Based on the nature of the detected attack, the following actions would be taken:

1. If the detected attack is simply a port scan or a probe, the HMM model attempts to make a prediction of a possible future attack based on the current distributed attack patterns. Based on this prediction, the central controller (or administrator) would take precautionary measures to prevent future attacks. The central controller would also make use of an online risk assessment of the assets subjected to this possible serious attack in the future.

2. If the detected attack is very serious, the central controller would take necessary actions to re-configure firewall rules or notify the administrator etc. Such serious attacks would bypass the HMM model.

3. At any time any abnormal traffic rate is noted by the monitor if a predetermined level is reached, the central controller may take necessary actions to re-configure firewall rules or notify the administrator etc.

In the DIPPS framework, each network component may host one or more IDS located in a distributed network. Since there will be a large number of flag generators, these must be abstracted, analyzed, and condensed by a suitable architecture before arriving at a final conclusion. Very often, it is to be noted that the event information, which is detected by the IDS agents will follow a bottom up approach for analysis and the various command and control flow will follow a top-down approach. The physical location of IDS agents may be fixed or mobile so as to monitor certain parts of the network segments.

The co-operative intelligent agent network is one of the most important components of the DIDS [1]. Ideally these agents will be located on separate network segments, and geographically separated. Communication among the agents is done utilizing TCP/IP sockets. Agent modules running on host machines are capable of data analysis and to formulate adequate response actions and may be implemented as read only and fragile. In the event of tampering or modification the agent reports to the server agent and automatically ends its life. Agents residing in the individual analyzer/controllers consist of modules responsible for agent regeneration, dispatch, updating and maintaining intrusion signatures and so on. These agents control the individual IDS agents for monitoring the network, manage all the communication and life cycle of the IDS agents and also update the IDS agents with detection algorithms as well as response and trace mechanisms.

3. Hidden Markov Model (HMM)

We model the interaction between the attackers and the system by a Markov model, and we assume the system to be in one of the following states; Normal (N) indicating that there is no ongoing suspicious activity, Intrusion Attempt (IA) indicating suspicious activity against the network, Intrusion in Progress (IP) indicating that one or more attacker have started an attack against the system, and Successfull Attack (SA) one or more attackers have already broken into the system. By using a Markov model, we assume that next state transition only depend on current state, this is known as the Markov assumption. To describe an IDS Agent we extend the Markov model to a Hidden Markov Model (HMM), by assuming that the alarms produced by the HMM Agent only depend on the state of the system. The word hidden indicates that the state of the system is not possible to observe, but only observations (output from the IDS Agents) that depend on the system state. Observations from the IDS Agents are used to estimate the system state distribution. One HMM model is used for each IDS Agent, and the state estimation is updated for each new observation from the IDS Agent. The state distribution is further used to estimate the intrusion frequency. The use of HMM to model the interaction between an attacker and a system is based on [2, 5], and [6] explain how to model the interacton between an attacker and a system using a Markov model.

4. Why Fuzzy Modeling?

If the problem to be solved can be described mathematically and there exist techniques to solve the problem by using reasonable computational power and time, this method should be preferred. But for some real world problems no solution is known at all, and for these problems heuristic techniques may be the only practical solution. An heuristic method is not guaranteed to give the best solution, but often gives a satisfying solution. One way to make a heuristic solution is to use previous experience and some general rules, this is a very natural approach for humans.

Risk assessment is often done by human experts, because there is no exact and mathematical solution to the problem. Usually the human reasoning and perception process cannot be expressed precisely. Different people have different opinions about risk and the association of its dependent variables, and fuzzy logic provides an excellent framework to model this. The key idea is to capture knowledge or information from risk managers and security experts and to embed this vital knowledge in the form of *if-then* rules in a fuzzy inference system to automate the risk assessment.

4.1. Fuzzy Modeling of Risk

The difference between an ordinary crisp set and a fuzzy set is that elements of a fuzzy set have a degree of membership. An element $(x,\ \mu_A(x))$ of a fuzzy set A is therefore a pair where $\mu_A(x)$ is a membership function and represents the degree of membership for x in A. The x value is called a crisp input, to indicate that it is a number. The membership functions for intersections and unions of fuzzy sets are normally constructed using the T-norm and the T-conorm operators. The most frequently used T-norm operator is $T_{\min}(a,b) = \min(a,b)$ and the most frequently used T-conorm operator is $T_{\max}(a,b) = \max(a,b)$.

The first step in the fuzzy inference system is to fuzzify the inputs, that is using the membership functions to calculate the degree of membership in different fuzzy sets. Next step is to apply *if-then* rules. For a Mamdani fuzzy system the *if-then* rules are of the form

$$if\ x\ is\ A\ and\ y\ is\ B\ then\ z\ =\ C \tag{1}$$

where A, B, C are fuzzy sets, and the first part (between if and then) is called the antecedent and the last part (after then) is called the consequent. Usually the T-norm and T-conorm operators are used in the evaluation of the antecedents and consequences respectively. After the *if-then* rules have been applied the crisp output is calculated through a process called defuzzification. But the most widely used defuzzificaton technique is possibly the centroid of an area:

$$Z_{COA} = \frac{\int_Z \mu_A(z) z dz}{\int_Z \mu_A(z) dz} \tag{2}$$

A unit consisting of fuzzification, rule evaluation and defuzzification is called a Fuzzy logic controller (FLC). In this paper we use a hierarchical structure where output from one FLC is used as input to another FLC.

For the risk assessment, nine basic linguistic variables are used that are processed using three Fuzzy Logic Controllers ($FLC_1 - FLC_3$). The three FLC's represent *Threat Level*, *Vulnerability* and *Asset Value*, which are three derived linguistic variables. The derived linguistic variables are then combined using FLC_4 to compute the net *Asset Risk*. This forms a hierarchical fuzzy system as shown in Figure 1. In this research, we used a Mamdani fuzzy inference system.

Values for the input variables are estimated based on the information from the HMM module, the DIDS and the traffic rate monitor. To simplify the calculation of input values, we have used the same attack categories as proposed by MIT Lincoln Laboratory - DARPA IDS evaluation datasets IDS [4]. The local controller uses information from the DIDS and the traffic rate monitor to predict which attack category the next attack will fit into.

The following sub-sections are strongly based on some of the principles of the FAIR method described in [3].

4.2. Fuzzy Modeling of Threat Level

Threat level is modeled using three linguistic variables: *intrusion frequency*, *probability of threat success* and *severity*. Three Membership Functions (MF) are used for the three inputs and the output variable.

Intrusion frequency describes the intensity of attack against the asset that is subject to monitoring. To estimate the intrusion frequency we use the output from the HMM module and count how often the probability of being in state *intrusion in progress* exceeds a specific limit. Intrusion frequency is measured as attacks/unit time.

Probability for threat success is estimated based on output from the DIDS, and describes how likely it is that an attacker will mange to overcome the proactive controls. The actual values are in the range $0 - 1$ and is stored in a lookup table.

Severity describes the impact of an attack on the asset.

All input variables to FLC_1 have three different linguistic values *Low*, *Medium* and *High*. The output from FLC_1 is Threat Level, and Fig 2 illustrates the *if-then* rules implemented in FLC_1 as a fuzzy associative memory (FAM). Figure 3 shows the controll surface view of FLC_1 plotting *Threat Level* as a function of *Probability of Threat Success* and *Intrusion Frequency*.

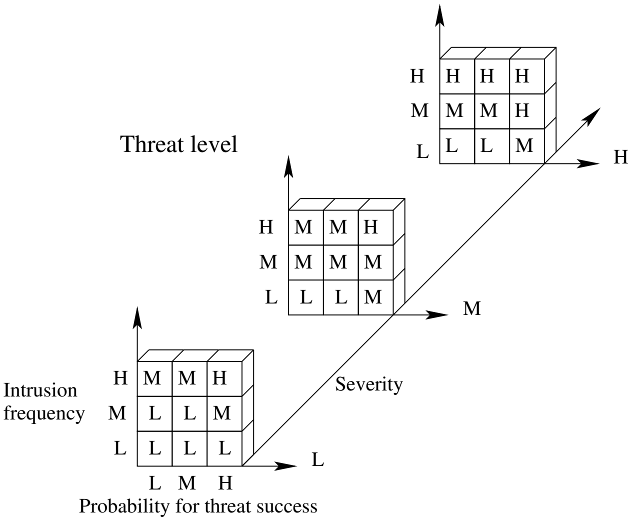


Figure 2. Sliced cube FAM representation of $\mathbf{FLC_1}$

4.3. Fuzzy modeling of Vulnerability

The Vulnerability is estimated in FLC_2. Vulnerability may be defined as the probability that an asset will be unable to resist the action of a threat agent [3]. In this paper we model vulnerability as a derived variable from *Threat Resistance* and *Threat Capability*. Three MF are assigned to each of the two input variables and the output variable.

Threat resistance is the strength of the security measures compared to the forces the attacker might use. One example of threat resistance is password length.

Threat capability is the level of force an attacker is capable of applying against an asset.

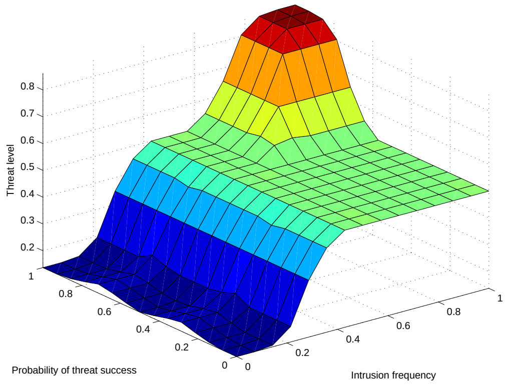
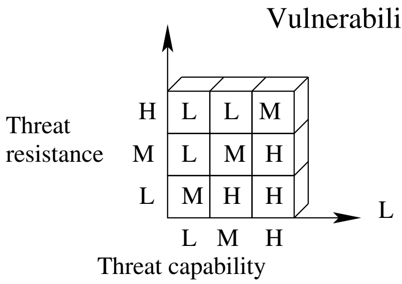


Figure 3. Controll Surface View of $\mathbf{FLC_1}$

The output variable from FLC_2 is the vulnerability and the *if-then* rules implemented in FLC_2 is depicted in Figure 4, and Figure 5 shows a control surface view of FLC_2.



Threat resistance	L	M	H
H	L	L	M
M	L	M	H
L	M	H	H

Figure 4. FAM representation of $\mathrm{FLC_2}$

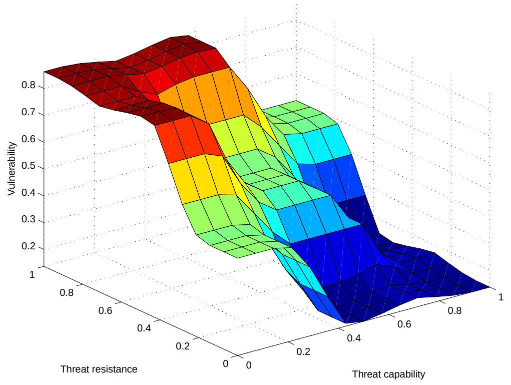


Figure 5. Control Surface View of $\mathrm{FLC_2}$

4.4. Fuzzy modeling of Asset Value

The asset value is estimated in FLC_3 and is derived from three linguistic variables: *Cost*, *Criticality*, *Sensitivity* and *Recovery*. An asset is any data, device or other component that supports information-related activities, and which can be affected in a manner that result in loss. For all the input variables of FLC_3, we used only two MF (to reduce the number of *if-then* rules needed). Three MF are used for the output variable. The *if-then* rules implemented in FLC_3 is shown in Table 1. The first four columns represents the input linguistic values: a the *Cost*, b *Criticality*, c *Sensitivity* and d *Recovery*. The last column labeled e represents the output variable *Asset Value*. A control surface view of FLC_3 is shown in Figure 6.

Cost (a) Represents the cost associated with an asset that have been stolen or destroyed

Criticality (b) Mainly characterizes the impact on an organization's productivity. This attribute is related to integrity and availability.

Sensitivity (c) Impact of confidential information being disclosed.

Recovery (d) How fast the loss can be re-stored and the asset be back to normal again.

Table 1. Rule table for $\mathrm{FLC_3}$.

	Innput				Output
Rule	a	b	c	d	e
1	L	L	L	L	L
2	H	L	L	L	H
3	L	H	L	L	M
4	H	H	L	L	H
5	L	L	H	L	M
6	H	L	H	L	H
7	L	H	H	L	H
8	H	H	H	L	H
9	L	L	L	H	L
10	H	L	L	H	H
11	L	H	L	H	H
12	H	H	L	H	H
13	L	L	H	H	H
14	H	L	H	H	H
15	L	H	H	H	H
16	H	H	H	H	H

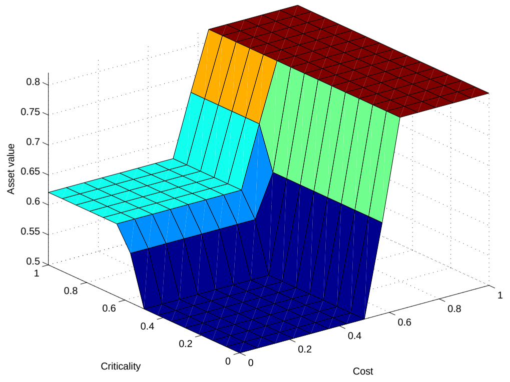


Figure 6. Surface plot of $\mathrm{FLC_3}$

4.5. Fuzzy Modeling of Risk

The risk is estimated by FLC_4 and is based on the output from the three fuzzy logic controllers $FLC_1 - FLC_3$.

For the input and output variables, three MF are used. The *if-then* rules used in FLC_4 is illustrated in Figure 7, and Figure 8 shows a control surface view of FLC_4.

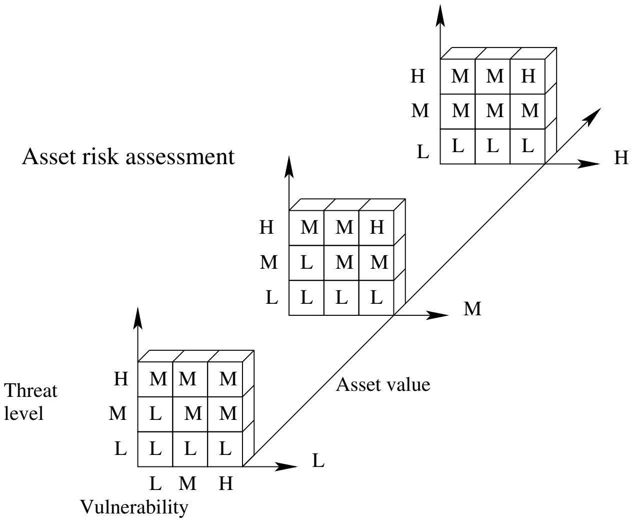


Figure 7. Sliced cube FAM representation of $\mathrm{FLC_4}$

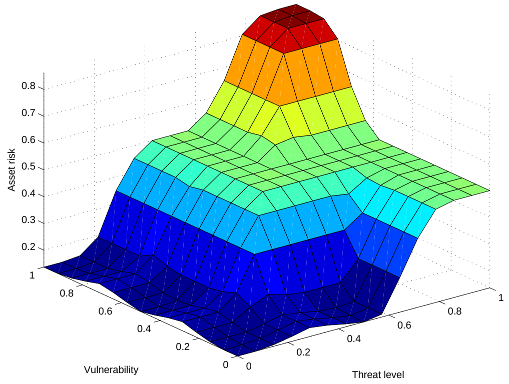


Figure 8. Surface plot of $\mathrm{FLC_4}$

4.6. Lookup Table

All the input variables except the Intrusion Frequency is decided based on the information received from the DIDS agents about the the ongoing attack, represented by a mapping $L(a) = y : A \to Y$ from attack types $a \in A = \{DoS, U2R, R2L, Pr\}$ to a parameter tuple $y \in Y = \mathbb{R}^8$. This function is implemented and referred to as a Lookup Table and contains parameters for different attack types. The information represents how vulnerable the system is to different attacks according to the value of different assets. These values have to be estimated by security experts.

For the risk assessment, two or three MF are proposed for each input variable, and three MF for the output variable. For two level input variables, we used the following two trapezoidal MF to define the *Low* and *High* linguistic values.

$$\begin{aligned} \mu_{L_2}(x) &= trap(x, -0.6, -0.2, 0.2, 0.6) \\ \mu_{H_2}(x) &= trap(x, 0.4, 0.8, 1.2, 1.6), \end{aligned} \tag{3}$$

For three level input and output variables, we propose two trapezoidal MF to define the *Low* and *High* linguistic values and a triangular MF to define *Medium* linguistic value (as illustrated below).

$$\begin{aligned} \mu_{L_3}(x) &= trap(x, -0.4, -0.1, 0.1, 0.4) \\ \mu_{M_3}(x) &= triang(x, 0.2, 0.5, 0.8) \\ \mu_{H_3}(x) &= trap(x, 0.6, 0.9, 1.0, 1.4). \end{aligned} \tag{4}$$

The membership functions used for input variables with three fuzzy sets are shown in Equation 5 and Figure 9.

All input variables are normalized and are members of the crisp set X defined as $X = \{x|0 \le x \le 1,\ x \in \mathbb{R}\}$. The parameterized MF used are triangular Equation 5 and the trapezoidal Equation 6.

$$triang(x, a, b, c) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \le x \le b \\ (c-x)/(c-b) & b \le x \le c \\ 0 & x > c \end{cases} \tag{5}$$

$$trap(x, a, b, c, d) = \begin{cases} 0 & x < a \\ (x-a)/(b-a) & a \le x \le b \\ 1 & b \le x \le c \\ (d-x)/(d-c) & c \le x \le d \\ 0 & x > d \end{cases} \tag{6}$$

All fuzzy *if-then* rules were formulated based on expert knowledge.

5. Experiment Results

To illustrate the risk assessment, we have created two lookup chart as shown in Tables 2 and 3. Specific information about different attack categories is stored in a lookup table. All values in the lookup table is scaled within the range $0 - 1$. The attack category used for the risk assessment is based on inputs from the IDS agents and this value is used to assign values to eight of the nine input variables. Only the Intrusion Frequency is estimated based on the output from the HMM module.

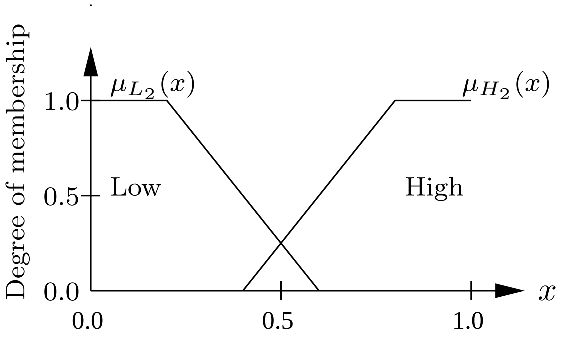


Figure 9. Two level membership function

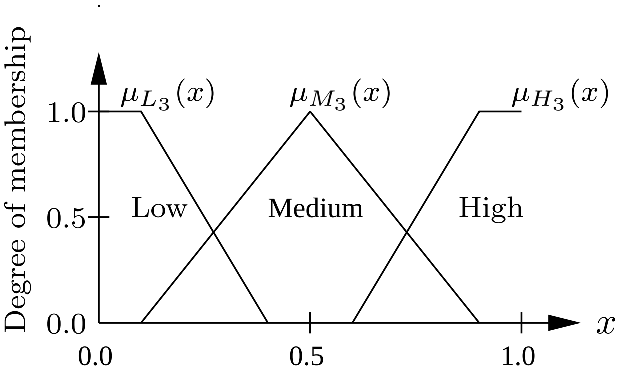


Figure 10. Three level membership function

Attacks are broadly divided into the following four categories: denial of service, remote to local, user to root and surveillance/probe.

A *denial of service* (DoS) attack is an attack where the attacker consume so much memory or CPU time that the legitimate users can not be served. Typical examples are Ping of Death, SYN Flood and Mailbomb. This attack is assumed to be easy to mount and indicated by high value for *Probability of Threat Success*. For DoS, the severity may be relatively low since it will not lead to much permanent damage. In most cases, the system may be restored to normal use once the attack is over.

An *User to root* (U2R) attack is an attack where an ordinary user in the system gain root access by exploiting some vulnerability in the system. Typical vulnerabilities that are exploited is buffer overflow and pure environment sanitation. A *remote to local* (R2L) attack is an attack where an attacker without an account on the computer tries to exploit some vulnerabilities to get access as an user of the computer. Possible attack strategies can be to exploit buffer overflows in network services software (imap, sendmail, apache). U2R and R2L categories are the most dangerous, since by gaining root access, the attacker could do almost everything with the system. Therefore a relatively high value for the *severity* is used in the above table. We assume the system to be well protected against U2R attacks indicated by relatively low *Probability of Threat Success*.

Table 2. Lookup Table1

	Attack Categories			
Variable	DoS	U2R	R2L	PR
Intrusion frequency	0.25	0.25	0.25	0.25
Pr threat success	0.90	0.70	0.70	0.10
Severity	0.40	0.90	0.90	0.30
Threat level	0.28	0.44	0.44	0.32
Threat resistance	0.10	0.60	0.90	0.20
Threat capabilit	0.50	0.85	0.80	0.10
Vulnerability	0.86	0.85	0.50	0.50
Cost	0.30	0.30	0.30	0.30
Criticality	0.70	0.70	0.70	0.10
Sensitivity	0.15	0.85	0.85	0.20
Recovery	0.40	0.85	0.70	0.15
Asset value	0.50	0.85	0.85	0.15
Asset risk	**0.34**	**0.50**	**0.50**	**0.40**

When an attacker uses some automated tools like Ipsweep, Nmap or Satan to gather information about the network and possible vulnerabilities we call it *probing*. This attack is assumed to be easy to mount and could pave way for further attacks.

Simulation results of the HMM is not reported in this paper due to space limitations, but the reader may consult [2] for some preliminary results.

Figure 11 illustrates the asset risk values for different intrusion frequency variations (0-1). For the different parameter settings (Tables 2 and 3), as evident from Figure 11, the asset risk values show clear sensitivity for each attack category. This also illustrates that the proposed system is very adaptive for different attack categories under varying conditions.

6. Conclusions

This paper proposed a detailed implementation of a fuzzy logic based online risk assessment scheme, which could aid the functioning of a Distributed Intrusion Prediction and Prevention System (DIPPS) for protecting high risk assets. The implementation of the proposed scheme is very simple and the developed system is easy to interpret. Our discussions with security experts and preliminary empirical results indicate that such a system is very practical for protecting assets, which are prone to severe attacks or misuse.

In the current fuzzy risk expert system, fuzzy *if-then*

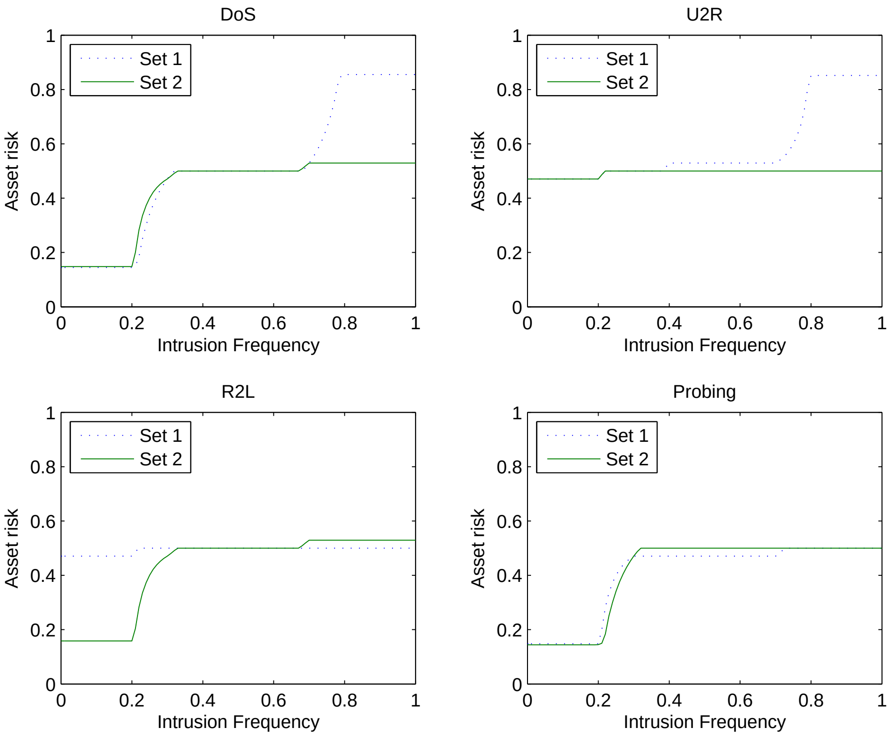


Figure 11. Parameter sensitivity for different attack categories

Table 3. Lookup Table 2

	Attack Categories			
Variable	DoS	U2R	R2L	PR
Intrusion frequency	0.25	0.25	0.25	0.25
Pr threat success	0.70	0.70	0.50	0.10
Severity	0.50	0.90	0.70	0.45
Threat level	0.32	0.44	0.32	0.28
Threat resistance	0.20	0.80	0.70	0.20
Threat capabilit	0.40	0.80	0.80	0.10
Vulnerability	0.85	0.50	0.63	0.50
Cost	0.40	0.40	0.50	0.30
Criticality	0.60	0.80	0.80	0.10
Sensitivity	0.20	0.80	0.70	0.10
Recovery	0.30	0.80	0.50	0.25
Asset value	0.50	0.84	0.82	0.15
Asset risk	**0.40**	**0.50**	**0.40**	**0.34**

rules were formulated based on expert knowledge. Our future research is targeted to develop adaptive fuzzy inference systems when some preliminary data or knowledge related to network risk is available. We also plan to investigate the use of different fuzzy inference methods.

References

[1] A. Abraham, R. Jain, J. Thomas, and S. Han. D-scids: Distributed soft computing intrusion detection systems. *Journal of Network and Computer Applications, Elsevier Science*, 30(1):81–98, 2007.

[2] K. Haslum, A. Abraham, and S. Knapskog. Dips: A framework for distributed intrusion prediction and prevention using hidden markov models and online fuzzy risk assessment. In *Third International Symposium on Information Assurance and Security, IEEE Computer Society press*, volume I, pages 183–188, 2007.

[3] J. Jones. An introduction to factor analysis of information risk (fair). *Norwich Journal of Information Assurance*, 2(1):67, 2006.

[4] K. Kendall. A database of computer attacks for the evaluation of intrusion detection systems. Master's thesis, MIT, USA, June 1999.

[5] R. Khanna and H. Liu. System approach to intrusion detection using hidden markov model. In *IWCMC '06: Proceeding of the 2006 international conference on Communications and mobile computing*, pages 349–354, New York, NY, USA, 2006. ACM Press.

[6] B. Madan, K. Vaidyanathan, and K. Trivedi. Modeling and quantification of security attributes of software systems. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'02)*, 2002.