HADMÉRNÖK

X. Évfolyam 1. szám - 2015. március

BEDERNA Zsolt
bederna.zsolt@bederna.hu

FUZZY-BASED INTRUSION DETECTION

Abstract

It is necessary to apply strong passwords on individual or distributed systems, to follow OS and application fixes and updates or even to make plans and implementations in a prudent way. But nowadays these are insufficient to reach expected security level. So, normally defendants follow a layered approach. One subset of those layers contains Intrusion Detection and Prevention System (IDPS). Unfortunately antecedents of IDPS can be foggy and therefore it may be hard to make clear decisions applying classic logic. Fuzzy logic can be helpful to handle uncertainty.

Manapság a megfelelő erősségű jelszavak használata, az operációs rendszer- és alkalmazásfrissítések követése, valamint a körültekintő tervezés, implementálás többnyire szükséges, de nem elégséges feltétele az elvárható biztonsági szint megteremtéséhez, így a védelemnek egy rétegzett struktúrát érdemes követnie. E struktúra egy részhalmaza az Intrusion Detection and Prevention System (IDPS) rendszerek. Mint oly sok esetben az informatika területén, az IDPS bemeneti adathalmaz elemeiről legtöbb esetben nem dönthető el a klasszikus logika szerint a célhalmazhoz tartozás. A fuzzy logika alkalmazásának létjogosultsága a kisebb-nagyobb bizonytalanság megléte.

Keywords: *intrusion detection, IDPS, fuzzy logic ~ behatolás-detektálás, IDPS, fuzzy logika*

INTRODUCTION

The interaction points between users and IT systems define attack vectors which are useful for penetrations. Attacks can be grouped so many way, e.g. there are inside and outside attacks or there are offline (physical) and online attacks. The last one has many subsets by services, applied protocols or even targeted OS or software versions. The followings are necessary tasks for a successful attack as EC-Council CEH [*1*]:

1. Reconnaissance,
2. Scanning,
3. Gaining Access,
4. Maintaining Access,
5. Covering tracks.

For us the most important task is scanning, which includes recognizing services and their vulnerabilities. As Figure 1 states, scanning has the aims of recognizing [*1*]

- Live systems and their open ports,
- Their services and
- The containing vulnerabilities.

Active scanning has some variables such as the number of attackers, the number of targets or even the timescale of the attacking procedure. When looking for parameters of the scanning tool Nmap, there are some possibility for controlling timescale of the corresponding task, e.g. scan-delay/max-scan-delay, min-rate/max-rate. A vulnerability scan can be attained manually or even automatically by a scanning tool such as Nessus[1].

Although the following publicly available databases do not contain zero-day vulnerabilities, but they could help in identification steps:

- CVE database[2],
- Microsoft Security Bulletin[3],
- The Open Source Vulnerability Database (OSVDB)[4].

The defending side is also doing its job in a more precise and sophisticated (or even a more complex) way than many years ago. Nowadays the integrated appliances can be the pioneers for defending networks. The less diversity is better economically and for manageability, performance and functionality. In the other hand nor a dedicated IDPS or an integrated appliance can offer complete solution. Protection against APTs (Advanced Persistent Threat) has more challenges, and it can be a single point of failure. Moreover it senses an untrue security feeling by an incorrect implementation or any malfunction during its operation. In 2012, an EC-Council trainer, Joe McCray pointed this fact in his presentation on Hacktivity [*2*].

BASICS OF IDPS

One of the objectives of an Intrusion Detection System is monitoring the usage of network and/or system resources. That is the root of recognizing violations of a security policy or a malware activity. Therefore it must be able to alarm the security personal or even other systems, and it must be able to hinder any further activity.

[1] http://www.tenable.com/
[2] http://cve.mitre.org/
[3] http://technet.microsoft.com/en-us/security/bulletin/
[4] http://www.osvdb.org/

Particularly an IDPS has the following parameters [*3*]:

- Reliability,
- Information gathering,
- Performance,
- Load handling or even load balancing,
- System and rule updates frequency,
- User-friendliness and manageability.

Components of IDPS

Typically an IDPS has the following components [*4*]:

- Agent/Sensor,
- Management server,
- Database server,
- Management console.

Agents and sensors are gathering real time information about monitored systems. Sometimes they can offer intervention to stop a task or a specific communication. Agents work inside hosts, while sensors work with networks.

Not every IDPSs has a management server function, but if it has, it can oversee the gathered data by agents and/or sensors. Making central decisions can be another important role as well as report generations. Nevertheless information must be stored somewhere and it must be seen somehow by a security administrator. These tasks are served by a database server and a management console.

Another special component of an IDPS is the security personal who must be trained and they must work with the implemented system and response to security incidents. The last task is usually a function of a Computer Security Incident Response Team (CSIRT).

Types of decision-making

Several methodologies are applied in IDPSs. Some of them use preset rules, while some of them employ decision-making on behavioral patterns. The basic ones are [*5*]:

1. Signature-based
    a) Pattern matching
    b) Stateful matching
2. Anomaly-based
    a) Statistical anomaly-based
    b) Protocol anomaly-based
    c) Traffic-anomaly-based
    d) Rule- or heuristic-based

Signature-based

As anti-virus solutions have predefined signatures, the IDPSs also have their own datasets containing similar data. The constant problem of this kind of defending approach is the deferred activity. Another problem can be the heavy dependence on vendor's continual updating services, and there can be wrong data included in datasets. In spite of these, signature-based defending systems are commonly implemented. The matter of fact they can be helpful for security engineers in basic tasks.

Anomaly-based

A profile-based methodology is applied to look after deviations against normal behavior. This may be used statistically or it can check traffic against protocol specifications and traffic patterns dynamically. The specification of normal behavior could be a huge problem.

Evaluation of IDPS

Measurement is very important to properly follow a system during its lifecycle. IDPSs do not exempt from this general truth. Many metrics can be defined, the followings are the most important for now:

- False alarm rate (FAR),
- False negative rate (FNR),
- Detection rate (DR).

FAR is a ratio where the number of false positive hits is divided by the sum of false positive and true negative hits. This sum is also known as the real activities. FNR is a ratio where the number of false negatives is divided by the sum of false negatives and true positives. This sum is the noxious activities. DR is that ratio where the number of true positives is divided by the number of noxious activities.

In connection with anomaly-based approach, integration of fuzzy logic into IDPSs can give us better efficiency by managing foggy and/or inadequate information.

BASICS OF FUZZY LOGIC

Fuzzy logic can be defined as a mathematical apparatus to handle inaccuracy caused by equivocality, uncertainty or lack of information.

The mentioned apparatus is based on fuzzy sets, fuzzy membership functions and fuzzy operators. Linguistic variables has been introduced by Zadeh. These variables can be any word or term corresponding to predefined fuzzy sets. In fact a membership function is the extension of characteristic function. While a characteristic function points $\{0,1\}$ for a membership, a corresponding fuzzy set points $[0,1]$ interval ($\mu_X: X \to [0,1]$). Needless to say, there are many membership functions like triangle, trapezoid and sigmoid (Figure 1). While triangle and trapezoid can be easily applied, sigmoid is excellence in complex tasks. If the sum of all membership value is one ($\forall i \forall x: \sum \mu_{A_i}(x) = 1$), the family set satisfies the condition of Ruspini partitioning. This can be easily reached by triangle and trapezoid functions.

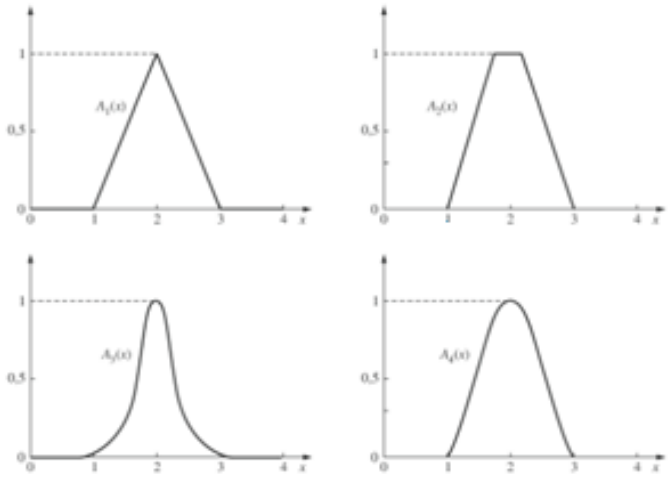

Figure 1. Fuzzy membership functions [*6*]

In fuzzy theorem, t-norm is a conjunction as c(x,y) and t-conorm is a disjunction as d(x,y). Both of them are dual operator. Notable operators are displayed in the following table.

Fuzzy operator	c(x,y)	d(x,y)
Min-Max	$min(x,y)$	$max(x,y)$
Product	xy	$x+y-xy$
Drastic	$\begin{cases} x, & y=1 \\ y, & x=1 \\ 0, & otherwise \end{cases}$	$\begin{cases} x, & y=0 \\ y, & x=0 \\ 1, & otherwise \end{cases}$
Hamacher	$\frac{xy}{\gamma+(1-\gamma)(x+y-xy)}$	$\frac{x+y-(1-\gamma')xy}{1+\gamma'xy}$
Einstein	$1-\frac{(1-x)+(1-y)}{1-(1-x)(1-y)}$	$\frac{x+y}{1+xy}$

Table 1. Characteristic fuzzy operators

Of course, negations play important role, too, but as the dual operators are extended for fuzzy logic, that must be done in case of negations (n: [0,1] → [0,1]). Basically $n\big(\mu(x)\big)=1-\mu(x)$ is applied, while there are others like Yager ($n\big(\mu(x)\big)=\sqrt[w]{1-\mu(x)^w}, w\epsilon[0,\infty[$) and Sugeno ($n\big(\mu(x)\big)=\frac{1-\mu(x)}{1+\alpha\mu(x)}, \alpha\epsilon[-1,\infty[$).

BASICS OF DECISIONS APPLYING FUZZY LOGIC

A fuzzy control has three basic steps:

1. Fuzzyfication,
2. Application of rules,
3. Defuzzyfication as needed.

In 1973 Zadeh created a fuzzy control model and in 1975 Mamdani created a less complex and easier applicable one equivalent with its predecessor. Later Takagi-Sugeno model was created as a seemingly different model, but their asymptotic equivalence relationship was proved by Dr. László Kóczy [*6*]. Mamdani model has a drawback as it may get into an instable state, while the stability of a system can be easier guaranteed by Takagi-Sugeno. Nevertheless the Tsukamoto control model may be also applied in control systems. The difference between the three models is displayed below.

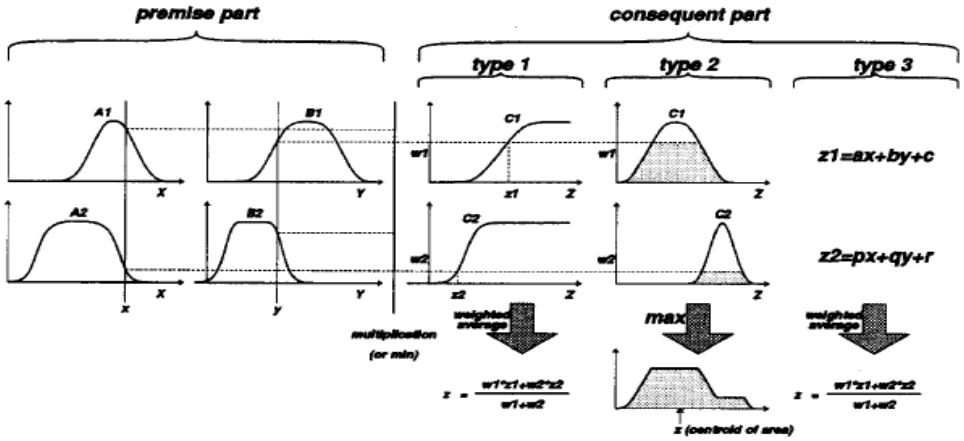

Figure 2. Generally used fuzzy control models5 [7]

Applied control and decision-making methods like association, clustering, decision tree can be extended for fuzzy theorem, too.

[5] Tsukamoto – type I., Mamdani – type II., Takagi-Sugeno – type III.

Fuzzy association

Association is an important element of data mining where a subset of information can be correlated with another subset. This kind of relationship has support and confidence. Support of $X \rightarrow Y$ rule is the percentage of transactions containing $X \cap Y$. Confidence of $X \rightarrow Y$ rule is the percentage of transactions containing $X \cup Y$. These metrics are specific for a particular set that can be explored by Apriori [*8*,*9*], FP-Growth [*8*,*9*,*10*], Fuzzy Grids-based Rules Mining Algorithm (FGBRMA) [*11*].

Fuzzy clustering

Basically clustering divides an X set to its subsets where items in each subset have much more similarities and two items connecting to different subsets are much more difference. In other words, partitioning an X set to c fixed number subsets is the aim. Application of classic logic gives strict membership for a particular subset, so an item can belong only one subset at once. Conversely fuzzy logic gives the chance for items to belong more subsets. The m parameter determines the level of cluster fuzziness.

As c is predefined, we can compare outcomes with different c values by validity indexes, e.g. Partition Coefficient, Separation Index or Classification Entropy. Further indexes are in [*12*].

Fuzzy decision tree

Applying fuzzy logic in breakdown rules gives the advantage of using more than one pattern. Trees may grow vertically (left side of Figure 3) or horizontally (right side of Figure 3). Problems may arise during breakdown as the corresponding tree may overreach the optimal state. Zenon A. Sovnowski et al. has defined C-Fuzzy decision tree based on classic logic and FCM clustering [*13*].

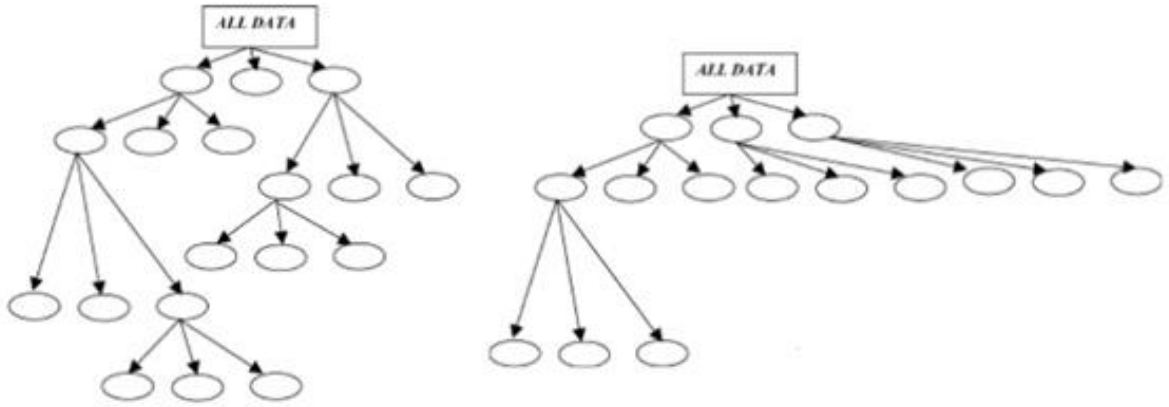

Figure 3. Growing possibility in decision trees [*13*]

EXPERIMENTAL SYSTEMS APPLYING FUZZY LOGIC

Applying Mamdani control

For a long time scan-detection was working less reliably. To handle this problem many solutions have been created, one of them was FB-Snort by Wassim El-Hajj et al. in UAE University [*14*]. As it was based on the open-source Snort, it was given name like its origin (FB is for Fuzzy Based). It applies a Mamdani control.

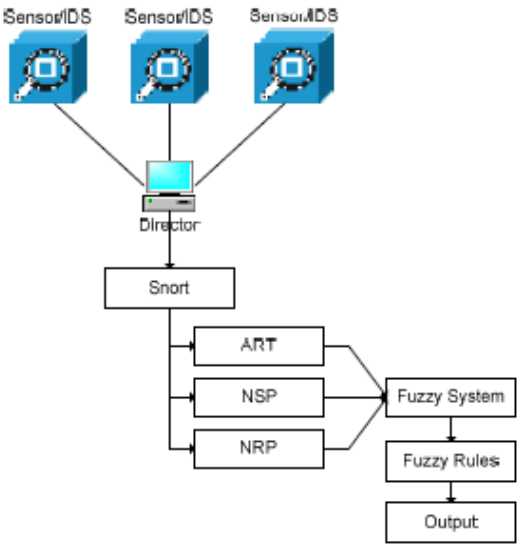

Figure 4. Architecture of FB-Snort [*14*]

Predefined parameters are:

- ART as average time between received packets,
- NSP as number of sent packets,
- NRP as number of received packets.

Figure 5 displays NRP which has partitions complying with Ruspini partitioning.

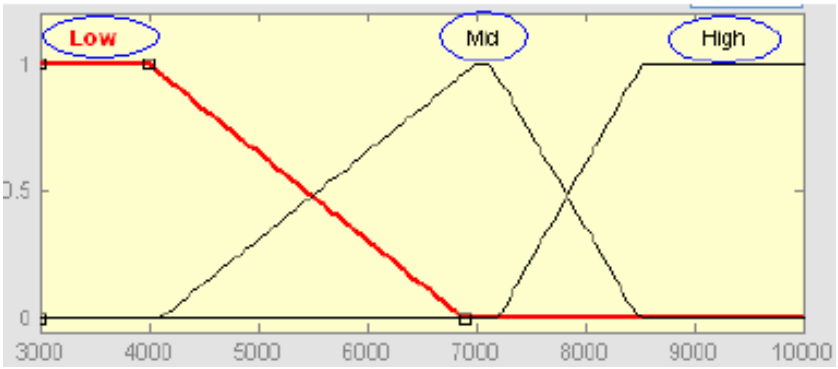

Figure 5. Fuzzyfication of NRP [*14*]

20 rules have been created from the maximum 27 rules, one of them is below. It represents the case when the average time between received packets is high and the number of sent packets is medium then the probability of our system is under pressure is high.

If (ART is high) and (NSP is med) and (NRP is high) then (output is high)

Compared to the conventional Snort, FB-Snort was able to decrease FAR while far more true positives could be reached.

Applying fuzzy clustering

Muna Mhammad T. Jawhar and Monica Mehrotra decided to create a fuzzy IDPS based on fuzzy clustering, as their expectation was to decrease FAR significantly by this way. The Fuzzy C-Means (FCM) clustering algorithm was chosen. It was followed by some kind of neural network [*15*].

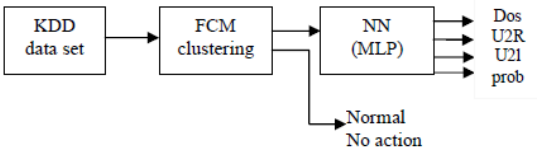

Figure 6. Block diagram of the hybrid fuzzy IDPS [*15*]

In connection with other studies, they chose KDD'99 dataset, too. From the 41 patterns of the dataset they utilized 35 ones to be processed by the FCM algorithm. FCM had c=2 (abnormal=1, normal=2) and m=2 parameters. They picked out randomly two subsets from KDD'99 for teaching purpose. After this, they reached DR=99.99% and FAR=0.0009%. Neural network was only used to distinguish attacks as Dos or Prob.

Attack name	FCM clustering			NN		
	Input	Output	Accuracy	Input	Output	Accuracy
Dos	23088	23089	99,90%	20463	20463	100%
U2R	7	7	100%	2	2	100%
U2L	608	608	100%	5	2	40%
Prob	1301	1301	100%	665	666	99,80%
Unknown	18	17	94,40%	114	166	68,60%

Table 2. Efficiency of FCM and NN [*15*]

Applying fuzzy association (FGBRMA)

In 2011, Mansour Sheikhan created a complex fuzzy IDPS. It was based on fuzzy association, genetic algorithm and fuzzy ARTMAP neural network [*16*]. FGBRMA was applied to recognize the association rules, to ascertain the optimal values of minimal support and minimal confidence. Its ARTMAP classification module was optimized by genetic algorithm. Finally it was used to check KDD'99 dataset, where it reached DR=97.11% and FAR=0.17%.

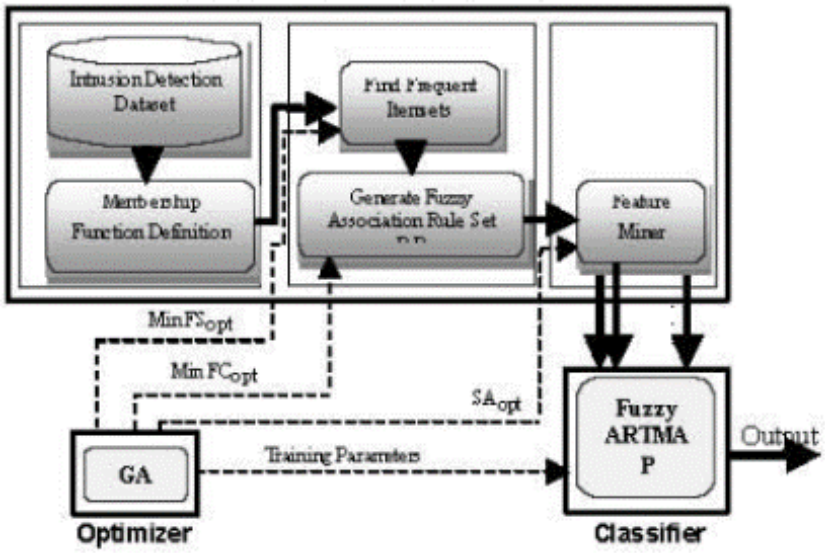

Figure 7. Hybrid IDPS of Mansour Sheikhan [*16*]

Applying fuzzy decision tree

In 2008 November, Krishnamoorthi Makkithaya et al. published their IDPS based on C-Fuzzy decision [*17*]. 34 parameters of KDD'99 dataset were taken into consideration. At first their tree size was 13 and their leaves number was 14. With this values they reached DR=94.84% and FAR=3.18%.

Second time they separated the beginning dataset to TCP, UDP and ICMP (Figure 8), as the parameters of this kind of decision trees are heavily influenced by the nature of the chosen clustering approach. With this step they reached DR=97.89% and FAR=0.99% for TCP, DR=98.32% and FAR=0.11% for UDP and DR=99.45% and FAR=19.44% for ICMP. In case of TCP the tree size was 9 and the leaves number was 10, in case of UDP the tree size was 7 and the leaves number was 8 and in the tree size was 5 and the leaves number was 6 for ICMP.

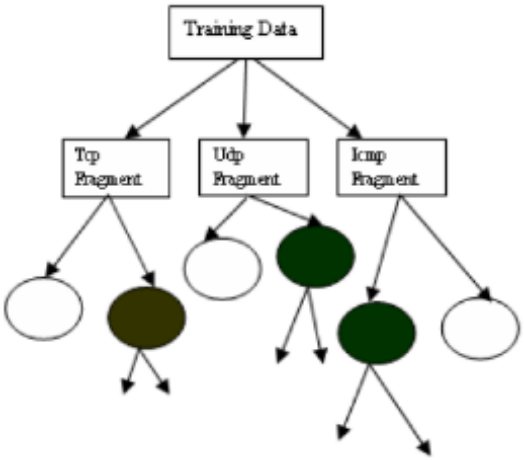

Figure 8. Modified horizontal clastering [*17*]

Mamdani control again

After all of these, I was interesting on employing a Mamdami control for a basic scan-detection. Main questions were for me:

- How easily can it be applied?
- What kind of efficiency can be reached?

As it was in Fuzzy control subsection, the first step is the fuzzyfication when parameters have to be defined. Contrary to FB-Snort, I took network investigation from network layer to transport layer. Therefore the main focus was on TCP and UDP. As there are big differences between them, in case of TCP I was on checking handshaking with *incompletedThreeWayHandshake* parameter, in case of UDP only the packet rate was watched with *packetRateUdp* parameter. Number of unused ports scanned by an attacker is playing an important role in both cases.

Parameters	TCP	UDP
Antecedents	incompletedThreeWayHandshake triedUnusedTcpPorts	packetRateUdp triedUnusedUdpPorts
Consequents	scanTcp	scanUdp

Table 3. Antecedent and consequent parameters

Fuzzy sets

Using markings displayed in Figure 9, the defined fuzzy sets are in the following two tables. For simple interpretation, the range was tightened to $[0, 100]$ for antecedents. Theoretically it has a range of $[0, +\infty[$, of course. For consequents, the interpretation range was $[0, 10]$.

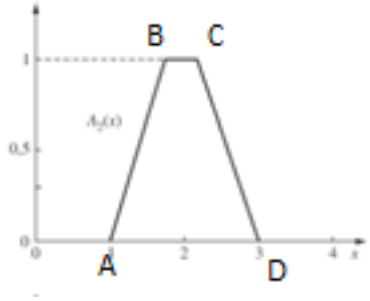

Figure 9. Trapezoid params

Variable	Fuzzy set	Point A	Point B	Point C	Point D
incompletedThreeWayHandshake	VeryLow	0	0	0,5	1,2
	Low	0,5	1,2	2,5	3,7
	Middle	2,5	3,7	4	5,8
	High	4	5,8	100	100
triedUnusedTcpPorts	VeryLow	0	0	0,5	1,4
	Low	0,5	1,4	1,9	3
	Middle	1,9	3	4	5,7
	High	4	5,7	100	100
scanTcp	VeryLow	0	0	0,5	1
	Low	0,5	1	2,5	3,5
	Middle	2,5	3,5	4	6
	High	4	6	10	10

Table 4. Fuzzy sets for TCP protocol

Variable	Fuzzy set	Point A	Point B	Point C	Point D
triedUnusedUdpPorts	VeryLow	0	0	0	1
	Low	0	1	1,5	2,7
	Middle	1,5	2,7	3,7	5,2
	High	3,7	5,2	100	100
packetRateUdp	VeryLow	0	0	0,5	1
	Low	0,5	1	1,5	3
	Middle	1,5	3	4	5
	High	4	5	100	100
scanUdp	VeryLow	0	0	0,5	1
	Low	0,5	1	2	3
	Middle	2	3	4	6
	High	4	6	10	10

Table 5. Fuzzy sets for UDP protocol

Inference rules

Avoiding instable states, I used all available 16 inference rules for both of TCP and UDP shown by the two tables below.

incompletedThreeWayHandshake	triedUnusedTcpPorts	scanTcp
VeryLow	VeryLow	VeryLow
VeryLow	Low	Middle
VeryLow	Middle	High
VeryLow	High	High
Low	VeryLow	Low
Low	Low	Middle
Low	Middle	Middle
Low	High	High
Middle	VeryLow	Middle
Middle	Low	Middle
Middle	Middle	High
Middle	High	High
High	VeryLow	Middle
High	Low	High
High	Middle	High
High	High	High

Table 6. Inferencing rules for TCP protocol

packetRateUdp	triedUnusedUdpPorts	scanUdp
VeryLow	VeryLow	VeryLow
VeryLow	Low	Middle
VeryLow	Middle	Middle
VeryLow	High	High
Low	VeryLow	Low
Low	Low	Middle
Low	Middle	High
Low	High	High
Middle	VeryLow	VeryLow
Middle	Low	Low
Middle	Middle	Middle
Middle	High	High
High	VeryLow	VeryLow
High	Low	Low
High	Middle	Middle
High	High	High

Table 7. Inferencing rules for UDP protocol

Test environment

Test environment was assembled with two virtual machines, one of them was serving SMB, RDP and Netbios, while the other one had moreover MSSQL.

Virtual Machine	TCP ports	UDP ports
#1	445, 3389	137
#2	445, 1433, 3389	137

Table 8. Test environment and its ports used by their services

Generating test data

The scanning device was a virtualized client machine and the scanning was made by Nmap with the following parameters:

Test #1
nmap -sT 172.17.5.250 -p 445,3389 -PN -T 5
nmap -sT 172.17.5.251 -p 445,3389 -PN -T 5
nmap -sU 172.17.5.250 -p 1434 -PN -T 5

Test #2
nmap -sT 172.17.5.250-251 -p 445,3389 -PN -T 1
nmap -sU 172.17.5.250 -p 1434 -PN -T 1

Test #3
nmap -sS 172.17.5.250-251 -p 445,3389 -PN -T 1
nmap -sU 172.17.5.250 -p 53,1434 -PN -T 1

Test #4
nmap -sX 172.17.5.250 -p 445 -PN -T 1

Test #5
nmap -sS 172.17.5.250-251 -p 80,443,445,3389,8443 -PN -T 1

Evaluation and further possibilities

It can be determined, this kind of control can be confidently recognize scans against unused ports. Nevertheless it gave a time independent solution (Test #1 and Test #2), albeit there was no chance to handle willfully wrong packets like in Xmas and Null scans (Test #4). This defect can be easily corrected by a *wrongFlagedPackets* parameter.

Test case	Consequent	VM #1	VM #2
Test #1	tcpScan	1,888888889	1,88888889
	udpScan	0,388888889	1,64285714
Test #2	tcpScan	4,0625	1,88888889
	udpScan	0,388888889	1,64285714
Test #3	tcpScan	2,902511467	1,88888889
	udpScan	0,388888889	1,64285714
Test #4	tcpScan	0,388888889	0,38888889
	udpScan	0,388888889	0,38888889
Test #5	tcpScan	7,466666667	7,46666667
	udpScan	0,388888889	0,38888889

Table 9. Results

SUMMARY

IT services are under pressure from the points of functionality and security. As it was discussed, while something seems to be black or white, 1 or 0 or even true or false, the most of the information we got are fuzzy. When I wrote about the applicability of fuzzy logic in IDPS in IT Business, their connection was confirmed by some vendor anonymously. Of course, no more information was given by them. This fact and the presented researches are pointing fuzzy logic is effecting our defending systems in great way.

References

[1] EC-Council, *Ethical Hacking and Countermeasures v6.1*.: EC-Council, 2010.

[2] Joe McCray, "Big Bang Theory: The Evolution of Pentesting High Security Environments," in *Hacktivity*, Budapest, 2012.

[3] ITServices, *IDS IPS Buyer's Guide*.: www.itservices.com, 2007.

[4] Karen Scarfone et al., *Guide to Intrustion Detection and Prevention Systems (IDPS)*.: NIST, 2007.

[5] Shon Harris, *CISSP® All-in-one Exam guide, 6th ed.*: McGraw-Hill Education, 2013.

[6] Kóczy, Tikk, *Fuzzy rendszerek*. Budapest: Typotex, 2001.

[7] Jyh-Shing Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man, and Cyberneticx*, pp. 665-685, May 1993.

[8] Lukas Helm, *Fuzzy Association Rules*.: Vienna University of Economics and Business Administration, 2007. [Online]. http://michael.hahsler.net/stud/done/helm/fuzzy_AR_helm.pdf

[9] Dr. Bodon Ferenc, *Adatbányászati algoritmusok*.: BME, 2010. [Online]. http://www.cs.bme.hu/~bodon/magyar/adatbanyaszat/tanulmany/adatbanyaszat.pdf

[10] Christian Borgelt, *An Implementation of the FP-growth Algorithm*., 2005. [Online]. http://www.borgelt.net/fpgrowth.html

[11] Yi-Chung Hu et al., "Discovering fuzzy association rules using fuzzy partition methods," *Knowledge-Based Systems*, vol. 16, pp. 137–147, 2003.

[12] Hoel Le Capitaine et al., "A cluster validity index combining an overlap measure and a separation measure based on fuzzy aggregation operators," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 3, pp. 580-588, 2011.

[13] A. Sosnowski et al., "C–Fuzzy Decision Trees," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART*, vol. 35, no. 4, pp. 498-511, Nov. 2005.

[14] Wassim El-Hajj et al. (2008) On Detecting Port Scanning using Fuzzy Based Intrusion Detection System. [Online]. http://www.academia.edu/1405670/On_detecting_port_scanning_using_fuzzy_based_intrusion_detection_system

[15] Muna Mhammad T. Jawhar et al., "Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network," *International Journal of Computer Science and Security*, vol. 4, no. 3, pp. 285-295, 2010.

[16] Mansour Sheikhan et al., "Intrusion Detection Improvement Using GA-Optimized Fuzzy Grids-Based Rule Mining Feature Selectror and Fuzzy ARTMAP Neural Network," *World Applied Sciensces Journal*, vol. 14, no. 5, pp. 772-781, 2011.

[17] Krishnamoorthi Makkithaya et al., "Intrusion Detection System using Modified C-Fuzzy Decision Tree Classifier," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 11, pp. 29-35, Nov. 2008.