CISJ

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

HIERARCHICAL CLUSTERING ALGORITHM FOR DETECTING ANOMALOUS PROFILES IN COMPUTER SYSTEMS

Rachid Beghdad

Faculty of sciences, 12 boulevard Bouaouina, Béjaïa 06000, Algeria.
Tel: (213) 34 22 33 14, Email: rbeghdad@yahoo.fr

Abstract: *We introduce a new intrusion detection method based on the Hierarchical Clustering Algorithm (HCA), to detect anomalous user's profiles. In the Unix system, a simple user has only some privileges (can access to some resources), but the root user has more privileges. So, we can speak here about hierarchy of users. By the same way, we can use a hierarchy of users in intrusion detection field, to distinguish between the normal user and suspicious user. Many data mining methods were already used in previous works in intrusion detection. Even if some of them led to interesting results, but they still suffer from some weaknesses. This is the reason why we focused in this study on the use of the HCA to detect anomalous profiles. A survey of intrusion detection methods is presented. The HCA procedure is described in detail. Our simulation results demonstrate the robustness of our approach in comparison to some previous used methods.*

Keywords: *Intrusion detection systems, Audit trail analysis, Hierarchical Clustering Algorithm, User behavior, Anomaly intrusion detection, Anomalous behavior.*

1. INTRODUCTION

An intrusion can be defined as a serie of activities aiming at compromising the security of a computer network system [1]. Intrusions may take many forms: external attacks, internal misuses, network-based attacks, information gathering, denial of service, and so on. Intrusion detection is an important step of protecting the computer network system from intrusions. Intrusion detection systems (IDS) are used to detect, identify and stop intruders. The administrators can rely on them to find out successful attacks and prevent a future use of known exploits. IDS are also considered as a complementary solution to firewall technology by recognizing attacks against the network that are missed by the firewall.

There are two basic types of intrusion detection: host-based and network-based. Each has a distinct approach to monitoring and securing data, and each has distinct advantages and disadvantages. In short, host-based IDSs examine data held on individual computers that serve as hosts, while network-based IDSs examine data exchanged between computers.

In addition to that, intrusion detection techniques can be mapped into four classes: anomaly detection, misuse detection, specification-based detection, and model-based detection. Anomaly detection consists of establishing normal behavior profile for user and system activity and observing significant deviations of actual user activity with respect to the established habitual pattern. Misuse detection, refers to intrusions that follow well defined attack patterns that exploit weaknesses in system and application software. In specification-based detection, the correct behaviors of critical objects are manually abstracted and crafted as security specifications, which are compared with the actual behavior of the objects. Intrusions, which usually cause object to behavior in an incorrect manner, can be detected without exact knowledge about them. Model-based intrusion detection compares a process's execution against a program model to detect intrusion attempts.

We introduce here an anomaly intrusion detection method based on HCA [2]. This method aims to find an optimum clustering (the best one) of users through a certain number of clusters *k* fixed a priori. If we find that some users are not well assigned according to this algorithm (they do not belong to their initial cluster), then, we can conclude that they are suspicious (intruders).

The rest of this paper is organized as follows. Section 2 presents a survey of some intrusion detection methods. Our approach based on HCA is detailed in section 3. Other clustering techniques are presented in section 4. Section 5 describes some experiments. Section 6 concludes the paper.

2. STATE OF ART

In this section, intrusion detection models are presented.

2.1. ANOMALY DETECTION MODELS

Learning Vector Quantization network. In [3] the authors described some preliminary results concerning the robustness and generalization capabilities of machine learning methods in creating user profiles based on the selection and subsequent classification of command line arguments. They based their method on the belief that legitimate users can be classified into categories based on the percentage of commands they use in a specified period. The hybrid approach they employed begins with the application of expert rules to reduce the dimensionality of the data, followed by an initial clustering of the data and subsequent refinement of the cluster locations using a competitive network called Learning Vector Quantization (LVQ). Since LVQ is a nearest neighbor classifier, and new record presented to the network that lies outside a specified distance is classified as a masquerader. Thus, this system does not require anomalous records to be included in the training set.

Network-based Intrusion Detection Using NNs. The authors of [4] presented an anomaly detection system that detects network-based attacks by carefully analyzing the network traffic data and alerting administrators to abnormal traffic trends. It has been shown that network traffic can be efficiently modelled using artificial neural networks. Therefore they used MLP neural networks to examine network traffic data. In their system, it becomes necessary to group network traffic together to present it to the neural network. For this purpose, they used self-organizing maps, as they have been shown to be effective in novelty detection, automated clustering, and visual organization.

K-means clustering model. We introduce in [5] an intrusion detection method based on the K-means (KM) clustering method to detect anomalous users' profiles. The main idea was to define *k* centroids, one for each cluster, such that each cluster represents a given user profile. These centroids should be placed as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate *k* new centroids as barycenters of the clusters resulting from the previous step. After we have these *k* new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the *k* centroids change their location step by step until no more changes are done. Even if this method led to interesting results, but it suffers from its false alarm rate.

2.2. MISUSE DETECTION MODELS

eXpert-BSM. The “eXpert-Base Security Module” (eXpert-BSM) [6] is a real time forward-reasoning expert- system that analyzes Sun Solaris audit trails. eXpert-BSM's knowledge base detects a wide range of specific and general forms of misuse, provides detailed reports and recommendations to the system operator, and has a low false-alarm rate. Suites of eXpert-BSMs may be deployed throughout a network, and their alarms managed, correlated, and acted on by remote or local subscribing security services, thus helping to address issues of decentralized management. Inside the host, eXpert-BSM is intended to operate as a true security daemon for host systems, consuming few CPU cycles and very little memory and secondary storage, according to its authors.

CAML. The Correlated Attack Modeling Language (CAML) [7] uses a modular approach, where a module represents an inference step and modules can be linked together to detect multistep scenarios. CAML is accompanied by a library of predicates, which functions as a vocabulary to describe the properties of system states and events. The concept of attack patterns is introduced to facilitate reuse of generic modules in the attack modeling process. CAML is used in a prototype implementation of a scenario recognition engine that consumes first-level security alerts in real time and produces reports that identify multistep attack scenarios discovered in the alert stream.

PCA model. Shyu and al. [8] proposed a method that used principal component analysis (PCA) in intrusion detection problem where the training data may be unsupervised. Assuming that anomalies can be treated as outliers, an intrusion predictive model is constructed from the major and minor principal components of normal instances. A measure of the difference of an anomaly from the normal instance is the distance in the principal component space.

Chronicles model. In [9], the authors proposed a multi-alarm misuse correlation component based on the chronicles formalism. Chronicles provide a high level declarative language and a recognition system that is used in other areas where dynamic systems are monitored. This formalism allows them to reduce the number of alarms shipped to the operator and enhances the quality of the diagnosis provided. They illustrated how chronicles might solve some of current intrusion detection issues like alarm overload, false positives and poor alarm semantics.

Bayesian networks model. Johansen and al. [10]

suggested a Bayesian system which would provide a solid mathematical foundation for simplifying a seemingly difficult and monstrous problem that today's Network Intrusion Detection Systems (NIDS) fail to solve. The Bayesian NIDS (BNIDS) should have the capability to differentiate between attacks and the normal network activity by comparing metrics of each network traffic sample. Finally, such a NIDS should prove to be easily extendable and run in real-time while simple to maintain.

Decision trees model. Abbes and al [11] show that an arbitrary definition of the domain search for signatures causes the generation of false negatives.To overcome this problem, they rely on a protocol analysis approach that leads to the construction of decision trees in the initial phase of the IDS deployment.The built tree is adaptative to the network traffic characteristics since the features chosen to split the tree ensure the highest reduction of entropy or the lowest Gini impurity.In addition, pattern matching operations are integrated inside decision tree.They are triggered after achieving light verifications and benefit from a refined domain search of signatures.

2.3 SPECIFICATION-BASED MODELS

Ning and al. Model. In [12], the authors described their on-going research on intrusion detection for mobile ad hoc networks. In particular, they employed specification-based techniques to monitor the ad hoc on-demand distance vector (AODV) routing protocol, a widely adopted ad hoc routing protocol. AODV is a reactive and stateless routing protocol that establishes routes only as desired by the source node. AODV is vulnerable to various kinds of attacks. The authors analyzed some of the vulnerabilities, specifically discussing attacks against AODV that manipulate the routing messages. They proposed a solution based on the specification-based intrusion detection technique to detect attacks on AODV. Their approach involves the use of finite state machines for specifying correct AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications. In addition, one additional field in the protocol message is proposed to enable the monitoring. They illustrated that their algorithm, which employs a tree data structure and a node coloring scheme, can effectively detect most of the serious attacks in real time and with minimum overhead.

2.4. MODEL-BASED MODELS

Dyck model. Dyck model [13] is an example of static binary code analysis model-based intrusion detection.

It is the first efficient statically-constructed context-sensitive model. This model specifies both the correct sequences of system calls that a program can generate and the stack changes occurring at function call sites. Experiments demonstrate that the Dyck model is an order of magnitude more precise than a context-insensitive finite state machine model. With null call squelching, a dynamic technique to bound cost, the Dyck model operates in time similar to the context-insensitive model.

2.5. CRITICS

(i) The IDS based on expert systems is a solution to a system intrusion problem, but it leads to some difficulties:

– The knowledge base of the expert system has to be always updated, which may lead to huge database.

– If the knowledge base is too large, then the inference engine might be too complex due to high number of rules to manage.

(ii) Neural networks (NNs) may also be a solution to such a problem, but they also present some difficulties:

– The behavior of a user may change from time to time, and some NNs can not deal with this. In this case, the NNs will fail to detect intruders.

(iii) Some existing languages for intrusion detection must be tested using realistic data in different operating systems: Linux, Solaris, or Windows NT, Sun, ... In addition to that, some known languages for intrusion detection are used only to detect known attacks (misuse detection). It will be interesting to study how these languages can be operable with an other language based on anomaly detection.

(iv) PCA may be a solution to intrusion detection problem. Even if, this method reduces the number of the original variables used, it leads to some difficulties:

– the exact value of the threshold (the distance) that determines if a user is suspicious or not, is not given.

– we cannot estimite the dicrimination between the used variables. In fact, the discriminating power ratio is not used.

This is the reason why our objective is to design an automatic tool based on the HCA method, in order to increase the security audit trail analysis efficiency.

3. THE "HCA" PROCEDURE

In the Unix system, a simple user has only some privileges (can access to some resources), but the root user has more privileges. So, we can speak here about hierarchy of users. By the same way, we can use a hierarchy of users in intrusion detection field, to distinguish between the normal user and suspicious user. This is the reason why we introduce here a hierarchical clustering technique.

Given a set of N items to be clustered, and an N*N distance (or similarity) matrix, the basic process of hierarchical clustering (defined by S.C. Johnson in 1967 [2]) is this:

First of all, start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain. Second, find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less. Third, compute distances (similarities) between the new cluster and each of the old clusters using the avearge-linkage clustering. We consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Formally the HCA is:

The algorithm is composed of the following steps:

1. Begin with the disjoint clustering having level L(0) = 0 and sequence number *m* = 0.

2. Find the least dissimilar pair of clusters in the current clustering, say pair (*r*), (*s*), according to:

$$d[(r),(s)] = \text{avg } d[(i),(j)] \qquad (1)$$

where: *i* and *j* stands for a pair of existing clusters, *r* and *s* stands for clusters, d[(r), (s)] refers to the distance between the 2 clusters *r* and *s*, avg d[(i),(j)] stands for the minimum average distance over all pairs of clusters in the current clustering.

3. Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering *m*. Set the level of this clustering to

$$L(m) = d[(r),(s)] \qquad (2)$$

4. Estimate the proximity matrix, D, by deleting the rows and columns corresponding to clusters (*r*) and (*s*) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (*r*,*s*) and old cluster (*k*) is defined in this way:

$$d[(k), (r,s)] = \text{avg } d[(k),(r)], d[(k),(s)] \qquad (3)$$

5. If all objects are in one cluster, stop. Else, go to step 2.

4. OTHER CLUSTERING TECHNIQUES

Among the most other existing clustering algorithms, we can find univariate clustering (UC), k-means, and canonical discriminant analysis (CDA) techniques. We have already used the k-means technique in [5], but, and according to our knowledge, both CDA and UC were not used before in intrusion detection.

Univariate clustering. The Univariate Clustering (UC) [14] procedure follows a simple and easy way to classify a given data set through a certain number of clusters. The main idea of UC is to optimally partition users of a given computer system, in homogeneous clusters, based on their description using a single quantitative variable. The quantitative variable here is an element of the profile vector P_k. Formally, UC consists in obtaining a partition minimizing the within-class variance (W):

$$W=(1/n) \sum k \sum i \in Pk \ (xi - \mu k)(xi - \mu k)t \qquad (4)$$

where: X= (x_1,…, x_n) represents a set of *n* independent variables (known users), P_k stands for the cluster *k*, μ_k stands for the arithmetic mean of each cluster *k*.

4.1. K-MEANS

The K-means (KM) [15] procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume *k* clusters) fixed a priori. The main idea is to define *k* centroids, one for each cluster. These centroids shoud be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate *k* new centroids as barycenters of the clusters resulting from the previous step. After we have these *k* new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the *k* centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J=\sum j=1..k \sum i=1..n \ ||xi(j) - cj||2 \qquad (5)$$

where $||x_i^{(j)} - c_j||^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

4.2. CANONICAL DISCRIMINANT ANALYSIS

The Canonical Discriminant Analysis (CDA) [15-16] approach functions by projecting user profiles onto a discriminant plane, and by deciding if a given user belongs really to a known group of users by measuring the distance between this user and the centre of gravity (the mean) of the group. We consider a population Ω on which *n* categorical predictors (also referred to independent variables) x_1, x_2, ..., x_n and one response variable (also referred to dependent variable) *Z* are observed. We denote by *k* the number of categories of *Z*. CDA can be understood as an *exploratory tool* to describe the dependence relations of the response variable on the given set of predictors in the observed sample of cases; the *k* categories of the response variable define a partition of the population Ω into *k* groups (ω_1 , ω_2 ,..., ω_k) and the *n* predictors are observed to characterize the typologies of cases within each group. At the same time, CDA analysis can be also used to define a *decision rule* for assigning a new case to one class on the basis of the observations of the given predictors in the so-called *learning sampl*e; a method such as test sample or cross-validation is considered to estimate the accuracy of the decision rule.

5. EXPERIMENTS

In a given university we achieve a serie of simulations where we consider 100 users assigned *randomly* to 3 clusters, according to their use of four commands: *ftp*, *latex, xdvi,* and *finger*. If a given user initially assigned to a given cluster *c*, still belongs to the same cluster after applying HCA, then this user is "normal". Else, (he/she) will be considered as "suspicious". The following table is an example of collected data representing the occurences of the cited commands per studied user:

Table 1. Example of a data matrix

Commands **Users (Observations)**	*ftp* (X1)	*latex* (X2)	*xdvi* (X3)	*finger* (X4)	**Cluster**
User1 (O1)	51	151	634	501	3
User2 (O2)	44	219	636	495	2
User3 (O3)	43	173	468	254	1
User4 (O4)	30	166	720	401	1
....	....	....	....	....	....
....	....	....	....	....	....
User100 (O100)	39	105	1259	258	2

We apply HCA, UC, KM and CDA methods to find suspicious users. Here also, the same reasoning made with HCA, will be used for UC, KM, and CDA. We assigned initially each user to a given cluster, and we look if (he/she) is assigned to the same cluster after execution of the considered clustering technique. The results of these simulations are the following:

Table 2: Comparison between HCA, UC, K-means, and CDA methods

Clustering Method **Criteria**	HCA	UC	KM	CDA
Anomaly score interval	**[0.69, 0.98]**	[0.59, 0.75]	[0.55, 0.74]	[0.41, 0.56]
Average detection rate	**84%**	65%	64%	47%

According to figure 1, HCA is better than both UC, K-means, and CDA clustering methods, in detecting intrusions. In all these experiments, there is one and only one case where UC and K-means are lightly better than HCA. In all the other cases, HCA has a detection rate which is higher than the three other clustering algorithms.

6. CONCLUSION

In one hand, today, there are many IDSs, and each IDS has its advantages and its weaknesses. In the other hand, it is often difficult to compare IDSs because they do not use the same metrics (criteria). Following the anomaly detection approach, we studied the use of the HCA method for modeling and detecting anomalous profiles. Our experiments show that the proposed approach is, in general, better than the UC, KM, and CDA methods.

Our paper introduced a survey of some intrusion detection methods. It presented our new approach based on the HCA method, which includes the following three steps : collecting informations (auditing the system), applying HCA technique, testing and deducing the intruders. To validate our approach, some experiments are presented. Our proposed solution for intrusion detection is very easy for implementation in any system having the audit mechanism.

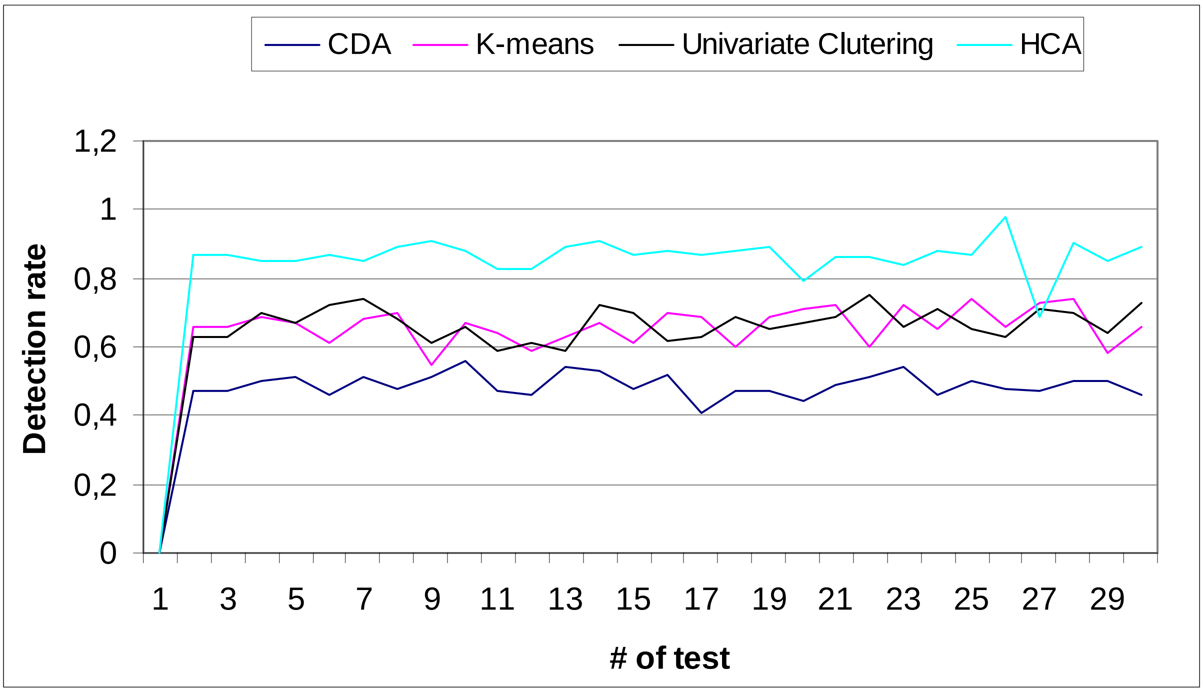

Figure 1: Efficiency of HCA method

7. REFERENCES

[1] 1. N. Ye, and X. Li, “A Scalable Clustering Technique for Intrusion Signature Recognition”, from the Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, U.S Military Academy, West Point, NY, 5-6 June, pp. 1-4, 2001.

[2] S. C. Johnson, "Hierarchical Clustering Schemes" *Psychometrika*, 2:241-254, 1967.

[3] J. Marin, D. Ragsdale, and J. Surdu, “A Hybrid Approach to the Profile Creation and Intrusion Detection”, technical report, Information Technology and Operations Center, United States Military Academy, 2000.

[4] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, M. Embrechts, “ Network-based Intrusion Detection Using Neural Networks”, technical report, Rensselaer Polytechnic Institute, Troy, New York 12180-3590, 2002.

[5] R. BEGHDAD, “K-Means for Modelling and Detecting Anomalous Profiles”, International Scientific Journal of Computing, volume 6, n°1, pp. 59-66, June 2007, Ukraine.

[6] U. Lindqvist and P. A. Porras, “eXpert-BSM: A Host-based Intrusion Detection Solution for Sun Solaris”, from Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001), pp. 240–251. IEEE Computer Society, New Orleans, Louisiana, 2001.

[7] S. Cheung, U. Lindquist and M. W. Fong, “Modeling Multistep Cyber Attacks for Scenario Recognition”, from the Third DARPA Information Survivability Conference and Exposition (DISCEX III), Volume I, pp. 284–292, Washington, D.C.2003.

[8] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. “A novel anomaly detection scheme based on principal component classifier”. In Proceedings of the Third IEEE International Conference on Data Mining (ICDM’03), pp. 172-179, Florida, Nov. 2003.

[9] B. Morin, H. Debar, “Correlation of Intrusion Symptoms : an Application of Chronicles”, In the Proceedings of the 6th Recent Advances in Intrusion Detection 2003 (RAID2003), 2003.

[10] K. Johansen and S. Lee, « CS424 Network Security: Bayesian Network Intrusion Detection (BNIDS), technical report, May 3, 2003.

[11] T. Abbes, A. Bouhoula, M. Rusinowitch, “Protocol Analysis in Intrusion Detection Using Decision Tree”, in the Proceedings of the International Conference on Information Technology Coding and Computing (ITCC’04), 2004.

[12] Peng Ning, Kun Sun, "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Adhoc Routing Protocols,". In Proceedings of the 4th Annual IEEE Information Assurance Workshop, pp. 60-67, West Point, June 2003.

[13] J. T. Giffin, S. Jha, B. P. Miller, ”Efficient Context-Sensitive Intrusion Detection”. In 11th Annual Network and Distributed Systems Security Symposium (NDSS), San Diego, California, February 2004.

[14] J. B. Mac Queen, ”Some Methods for classification and Analysis of Multivariate Observations”, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability"*, Berkeley, University of California Press, 1:281-297, 1967.

[15] G. J. McLachlan, “Discriminant Analysis and Statistical Pattern Recognition”, John Wiley & Sons, N.Y, 1992.

[16] L. Fisher, J. W. Van Ness, “Admissible Discriminant Analysis”, Journal of American Statistical Association, 68, pp. 603-607, 1973.

Rachid BEGHDAD *received his computer science engineer degree in 1991 from the Polytechnical school of engineers, Algiers, Algeria. He received his Master computer science degree from Clermont-Ferrand University, France, in 1994. He earned his Ph.D. computer science degree from Toulouse University, France, in 1997.*

He is a reviewer for some journals, such as the Computer Communications journal, Elsevier, UK.

His main current interest is in the area of computer communication systems including intrusion detection methods, unicast and multicast routing protocols, real-time protocols, and wireless LAN protocols.