Vert, G., Doursat, R. & Nasser, S. (2006) Towards utilizing fuzzy self-organizing taxonomies to identify attacks on computer systems and adaptively respond. *2006 IEEE World Congress on Computational Intelligence (WCCI/FUZZ-IEEE 2006)*, July 16-21, 2006, Vancouver, Canada.

Towards Utilizing Fuzzy Self-Organizing Taxonomies to Identify Attacks on Computer Systems and Adaptively Respond

Gregory Vert, René Doursat, and Sara Nasser

Department of Computer Science
University of Nevada, Reno
Reno, NV 89557, USA
gvert@cse.unr.edu

Abstract: - **Several methods for doing intrusion detection have been developed over the years. However, most of these methods are based on crisp statistical techniques that measure deviation from a norm. Due to the wide range of attacks on computers, statistical methods are not always effective because they aggregate many system variables into a single mathematical measure. Instead, taxonomies of attack features based on the concepts of fuzzy logic can be utilized to classify attacks and build simple response rules based on local system variables. Taxonomies however require correct hierarchial construction from subtaxonomies of atttack classifiers. An architecture that defines self organizing taxonomies based on fuzzy logic is therefore developed for future investigation.**

Key-Words: - Fuzzy Logic, Computer Security, Attacks, Taxonomy Systems

I. Introduction

The networking of computers has created an unprecedented opportunity for individuals to attack and destroy computer systems. It was recently estimated that about 10-20 new viruses appear daily [1] and, within any institution, security breaches are also reported almost daily.

Information resources are available that will notify users of new security holes [2,3,4,5]. Additionally, there are also a variety of security databases that will let users discover vulnerabilities associated with various software packages [6,7]. Several companies such as CERT keep large databases of known attacks [6,7,11]. Symantec has over 50,000 entries for known Internet security related threats [13]. With the proliferation of new attacks almost hourly, these databases can soon become unwieldy.

II. Problem Formulation

One approach to the classification problem is to develop a taxonomy of current attacks that classifies the various attack methodologies into distinct categories. By categorizing attacks, we can begin to look for patterns and common features of attacks. Standard responses to each attack classification can then be developed. This has the potential to possibly prevent new, unreported attacks from succeeding even without the installation of a patch. This approach must also consider how taxonomies are constructed. An approach is to construct a master taxonomy from subtaxonomies. However, construction of the master taxonomy must have a basis for its morphology.

There has been research attempting to classify different types of attacks, from Unix specific vulnerabilities [8,9] to network attack assessment [10]. This research has been important and useful, but its classification has focused on a specific class of attacks. We propose the classification of a broad range of computing attacks into a common hierarchy. This paper presents a novel approach to attack detection and defense that can potentially handle attacks by organizing them into taxonomic categories. Because attacks can often be similar in modality but require different responses, some attacks can be classified into different branches of the taxonomy. To solve this problem, we utilize fuzzy logic and fuzzy linguistic variable techniques to select an attack response. A method of developing standard responses to each attack classification is then developed. We further develop a method for determining how to construct a master taxonomy from subtaxonomies. This work has the potential to be highly beneficial to the security community.

III. problem solution

Attack databases, such as SecurityFocus's Vulnerability Database [6] and NIST's ICAT [7], list information about reported attacks, but they do not provide the means for dynamic classification of unknown attacks. In contrast, our approach describes a methodology that can potentially be used to identify known attacks and subsequently classify newly developed ones in real-time with the use of a taxonomy.

Our approach makes use of a set of attack attributes which describe how an attack executes. The attack attributes are populated with an attack's properties, which are then applied to an attack taxonomy for classification.

Attacks with common or similar attributes will be located in the same category of the taxonomic tree. By using this classification, a system could potentially be devised to take an appropriate action based on the classification of the attack within the taxonomies. This method can preclude a lengthy search through a large database of attacks for a possible defense.

Through careful choice of attribute list members, our taxonomy can conceivably support all known types of attacks. This attribute list can be altered in the future as new attacks present themselves.

A. Attack Attributes

We have developed and continue to develop a list of attributes to describe an attack and relate them in a fashion that would support taxonomic trees. We devised a top-down naming scheme. The first attribute is the root tree node and each subsequent attribute is a subnode. These attributes are period-delimited. For example, the Bandwidth attribute in the network taxonomy is written as:

Network.Bandwidth

and the InPorts under TCP in the same network taxonomy (Fig. 1) is written as:

Network.Protocol.TCP.InPorts

Our taxonomy includes 22 separate attack attributes. For the sake of brevity this list is not included in this paper. The attack attributes list a distinguishing set of actions and states of different attacks. We looked at different databases that compile information on existing attacks [6,7] and found the following general classifications among them:

- Remote access through a network connection
- Attacks that modify/create files on the file system
- Attacks executed using an exploit for a particular operating system and daemon
- Attacks that use kernel services for elevated privileges

Additionally we found that the abstract components of a computational system could be utilized in classification of exploits such as:

- physicality
- resources
- memory

B. Node Actionable Response Rules

Complex responses to an attack behavior can be created through the composition and action of a series of smaller, simpler rules. In this sense, we have placed into the nodes of our taxonomy simple localized rules that react to an attack on just the node characteristics. For example a rule in the Network taxonomy might be

Network.TCP.InPort = **n**
Network.ActionRule = **block n**

These are referred to as "node actionable response rules" and are collected as processing drops through layers of the taxonomies until it reaches a leaf node. There they form a complex set of responses to an attack which is actioned upon through the use of fuzzy logic.

C. Attack Taxonomies

In previous work several types of taxonomies were developed. The first type of taxonomy developed was based on common attributes of attacks that originated through a remote connection across a network.

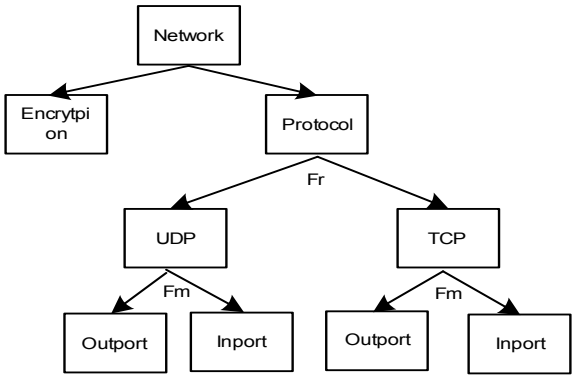


Fig. 1. Network Attribute Taxonomy

Fig. 1 presents the *network attribute taxonomy*. In our hierarchy, child nodes inherit all the attributes and descriptive properties of their parent nodes, as well as having node specific attributes. The network attributes specified in the tree help to define attacks based on the protocol, bandwidth, and action characteristics of the attack.

We also found an entire category of attacks on files and file systems as mentioned above. The *file system taxonomy* (Fig. 2) was developed to structure and organize this data into a taxonomic model. The attributes in this tree define what files on the victim's machine are created, changed, and deleted. It also allows for operating system specific attributes, such as registry entries in the Microsoft Windows environment.

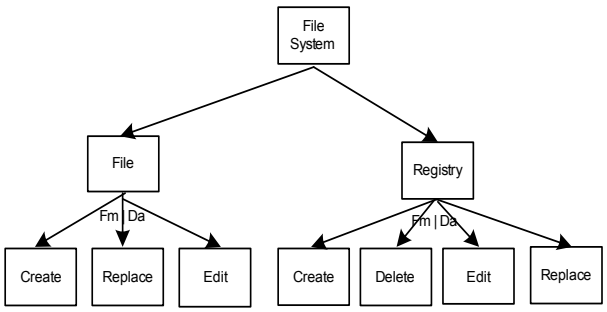


Fig. 2. File System Attribute Taxonomy

Another category of attacks are based on system exploits. Fig. 3 presents the *exploit attribute taxonomy* which was referenced in section III.A. The exploit tree defines the vulnerability that an attacker may use on a victim's machine. This taxonomy models common programming errors, improper configurations, and user errors.

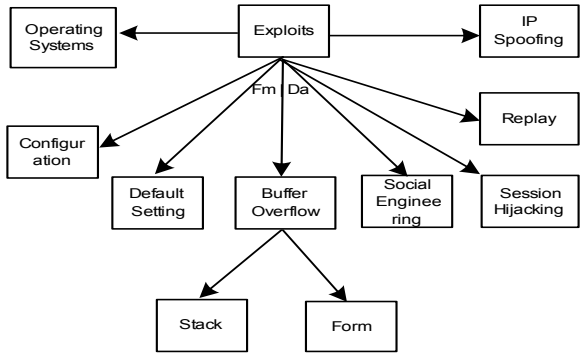


Fig. 3. Exploit Attribute Taxonomy

Finally, attacks exist that use services and drivers to gain elevated system privileges [14]. The *kernel taxonomy*, mentioned above, is presented in Fig. 4 and shows the types of attacks that are possible using kernel privileges.

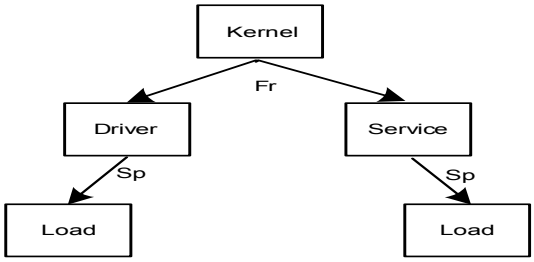


Fig. 4. Kernel Attribute Taxonomy

Newer attacks include the use of device drivers and kernel services that allow malicious users to completely bypass security and take complete control of the victim's computer.

We have defined in this previous work a preliminary set of attributes that populate the taxonomic subtrees and are utilized in the classification process. However further investigation has lead to the realization that attacks fall into several distinct categorical classifications based on the generalized subsystems found in a computer. These are now defined to be: network, file, memory, resource, kernel and physical systems. In addition, we needed to keep our previously defined exploit taxonomy but parse it into exploits based on what system is specifically attacked by the exploits. We have further developed the concept of memory-based, physical-based and network taxonomies, and exploits.

The first of our new taxonomic subtrees is based on physical access to a machine and is defined in Fig 5.

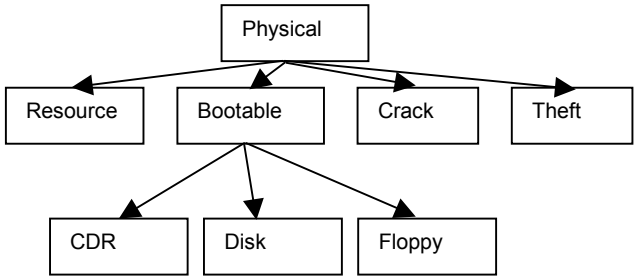


Fig. 5. Physical Attribute Taxonomy

This taxonomy classifies for attack attributes that are a physical characteristic in an attack. The next new taxonomic subtree we have developed bases classification on memory aspects of an attack and is shown in Fig. 6.

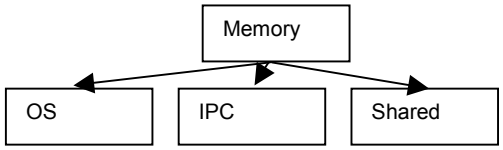


Fig. 6. Memory Attribute Taxonomy

The final taxonomic subtree we have defined in this phase of research is the resource subtree. This helps classify machine information based on resources being utilized. This tree is shown in Fig. 7.

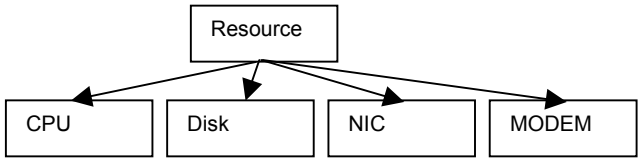
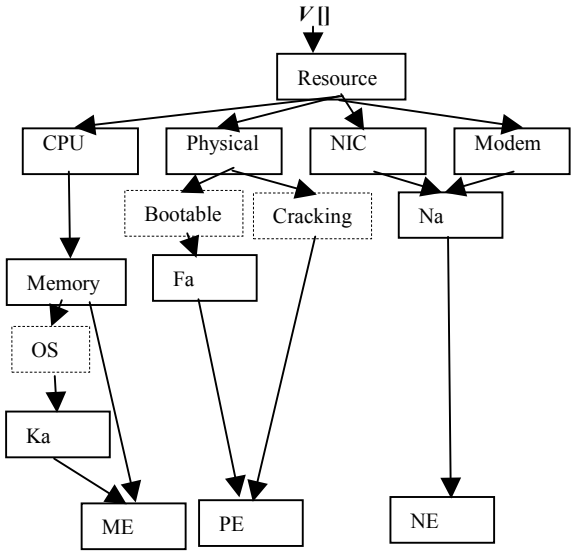


Fig. 7. Resource Attribute Taxonomy

Consolidating all of the above attacks into a single taxonomy produces a modified version of our earlier preliminary developed tree. This taxonomy is shown in Fig. 8, where each box represents the subtaxonomies presented in Fig. 1 through Fig. 7.



Ka - Kernel attribute tree
Na - Network attribute tree
Fa - File attribute tree
NE - Network-based Exploits
PE - Physical-based Exploits
ME - Memory-based Exploits

Fig. 8. Consolidated Taxonomic Graph

In Fig. 8 all leaf nodes are connected to the next subtaxonomic tree's root, except for boxes that have a dashed line. These boxes have child node hooks to a subtree that other child nodes to not follow through.

Input vector $\boldsymbol{V}$ is an n-dimensional feature vector whose attributes describe data about an attack as it is being observed. This vector contains the same attributes as those used in the subtrees when selecting and moving to the next child node. At this point in the development of our research it was realized that an attack can actually branch to two or more child nodes in a subtree or two or more subtrees in the consolidated taxonomic graph. The reason is that attacks are typically multi-pronged in their approaches. For instance, an attack may occur over the network primarily, however the instigator of an attack my also be sitting at a computer on the system trying to crack a password and gain physical access. For this reason, there may be multiple child nodes toward which an attack description can eventually bifurcate. However, attacks are typically going to have a preferred modality, e.g., Attack X primarily likes to use the network. For this reason, fuzzy logic was used to extend the above trees using linguistic variables and concepts of fuzzy object-oriented model design.

Fuzzy linguistic variables model the vagueness of human speech into a computable model. There are several approaches to this type of modeling [17]. One of the first tasks is to determine a suitable descriptive domain. Upon examination of our model we realized that the following domains would probably best describe the properties of an attack:

Frequency (*Fr*) = { *never*[0], *sometimes*[.25], *usually*[.5], *most of the time*[.75], *always*[1] }

Damage (*Da*) = { *none*[0], *unknown*[.30], *probable*[.55], *definite*[.80], *severe*[1] }

Speed (*Sp*) = { *none*[0], *below average*[.25], *average*[.5], *above average*[.75], *fast*[1] }

Familiarity (*Fm*) = { *known*[0], *similar*[.5], *unknown*[1] }

This suggests a classification tuple of fuzzy linguistic variables (FLV) where:

$$FLV[] = (Fr, Da, Sp, Fm).$$

The linguistic variables are shown where they are located in the taxonomy trees (Fig. 1 to Fig. 4) using their abbreviations mentioned above. At this phase of the research we has not defined the linguistic variables for the new taxonomies that have been developed. Each of the fuzzy linguistic variables are in the range [0, 1] where 0 and 1 are crisp. Fuzzy values associated with the variables are indicated above inside the brackets []. In addition to the input vector $\boldsymbol{V}$[] of characterizing attributes, we utilize fuzzy linguistic variables to characterize the attack. As input data in $\boldsymbol{V}$[] is classified and processed down through the tree, branch points of the taxonomy tree have the values for the linguistic variables assigned automatically as additional fuzzy characterizations of the attack. Selection of the correct fuzzy linguistic variables can be done by the system. This can produce a human-readable version of what the system thinks is happening. For example collection of data from computers currently being attacked may indicate that 75% of the time, a TCP port is selected for an attempted entry into the system. Considering that this is a frequency variable (*Fr*), the fuzzy values assigned

to the tcp attribue in $V[]$ might look like the following in the Na taxonomy tree:

Network.tcp = **most of the time** (fuzzy *Fr* = .75)
Network.udp = **sometimes** (fuzzy *Fr* = .25)
Network.TCP.InPort = **n**
Network.TCP.OutPort = **null**
Network.ActionRule = **block n**

Notice that node action rules are also found at each node in the subtrees and tailored to a localized response to the attack. However, they are not actioned until processing enters a leaf node, where they form a complex rule base tailored to the elements of the attack. This borrows from complex systems theory that supports the idea that a composite collection of small simple rules can from complex behaviors.

Once at the leaf nodes, where the set of all accumulated response rules are actioned, the values of the fuzzy classifiers are joined together through the following operation:

$$FAct = \frac{\sum_{i}^{n} \sum_{ii}^{|FLV|} FLV_i[ii] w_{ii}}{\sum_{i}^{n} \sum_{ii}^{|FLV|} FLV_i[ii]} \qquad (1)$$

where:
|FLV| - cardinality of the FLV vector
n - number of leaf nodes with *FAct* values.

Fuzzy actionability (*FAct*) values can exist in several leaf nodes ranging from large to small values. This borrows from the fact that an attack's classification mentioned earlier may go down multiple branches of a taxonomic tree based on how the *FLV*[] set is applied to attributes at each node. The concept is the same as the one found in fuzzy object-oriented diagrams and fuzzy subsets.

Fig. 9 illustrates this point. In this case an attack can crisply belong to an Ea leaf node, or a Ka leaf node. However, with the application of the *FLV* variables, it is possible that an attack belongs to one or more leaf nodes.

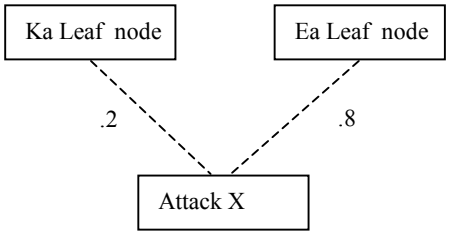


Fig. 9. Partial membership of attack in multiple leaf nodes

Fuzzy membership implies that an attack is a subset of a node by the following

$$A \subseteq B \quad \Leftrightarrow \quad \forall x \in U,\ u_A(x) \leq u_B(x) \qquad (2)$$

where:

U – all possible attacks
x – any attribute in A's attack vector $V[]$
A – set of attributes of vector $V[]$ for an attack X
B – set of attributes of vector $V[]$ for leaf nodes B.

The application of the response rules examines the *FAct* values and applies them in the following algorithm:

While (attack in progress)
build $V[]$
process taxonomic graph
node to action (NTA) = max[all leaf nodes]
set NTA.*FAct* = null
execute actionable rule set
End While

As an example of this algorithm, a reconnaissance attack to gather information might perform port scans on TCP or UDP ports. A potential response to this attack via *FAct* and action rules could be to deny access to the originator of the port scan. The system can optionally insert a firewall rule that blocks all future traffic from the attacker. For preventive measures, the firewall can be configured to deny all traffic and only allow packets from pre-determined static IP addresses [15]. There are also known methods that can be used to thwart OS fingerprinting techniques [16].

IV. SELF ORGANIZING TAXONOMIES

In previous work we made an initial attempt to develop the concept of attack taxonomies. The question of taxonomic structure was raised for further investigation. Specifically, how does one know that a subtaxonomy goes above or below another taxonomy when building the consolidated taxonomic graph seen in Fig. 8. It would be ideal if subtaxonomies could somehow suggest how to organize themselves or provide clues on how they should be organized.

We started to investigate this issue and had several initial observations. The first of these is that as a taxonomy increases in depth, its detail must also increase. Children nodes in a taxonomy inherit all the attribute classifiers that their parent has but also add their own. This can be seen in Fig. 10.

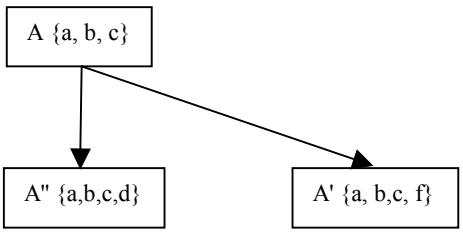


Fig. 10. Inheritance in child taxonomies

Fig. 10, A" and A' inherit attribute classifiers {a, b, c} from A, but differentiate themselves by classifiers d and f. Therefore:

$$A \subseteq A' \text{ and } A''$$

and

$$A' \neq A''.$$

Additionally

$$A' \subseteq A'' \text{ and } A'' \subseteq A'.$$

Utilizing these observations about relationships in taxonomies it is possible to determine the hierarchial relationship that subtaxonomies should use when being assembled into larger trees. In current research we extend this initial observation with two more concepts; that of cardinality and similiarity.

Cardinality in its simplest sense means that the number of elements in one set are the same as the number of elements in another set. For this concept we introduce the notation:

$$|A| = \{x : x \text{ is a natural number}\}$$

where:

$|A|$ represents the cardinality of feature set A.

The addition of cardinality to our model leads to the suggestion that we classify depth of nodes in our taxononomy by cardinality. However, this is not sufficient when considering how to determine if two or more nodes have the same parent node. In other words, two feature sets A, A" may have the same depth in the taxonomy but not be in the same branch of a subtree because they are not related. Therefore, we introduce the concept of similiarity. The determination of how to measure similiarity is left to future research but we can define the concept to be component of fuzzy linguistic variable of the form R for "relation" where :

nr – not related = {0}
sr – somewhat related = {.5}
or – often related = {.75}
un – unknown = {0}
re – related = {1}

We further defined the following sets of relations:

R = { sr, or, re}
R" = { nr, un}

where:

R is the set of related nodes
R" is the set of unrelated nodes.

The concepts of node relation and node cardinality can be utilized to create a self organizing taxonomy using the following algorithm:

given: R, R" and feature vector nodes A, A'
if A, A' $\in$ R
  if $|A| > |A'|$ then
    A is child of A'
    $A_l = A'_l + 1$
  else
    A' is child of A
    $A'_l = A_l + 1$

where:

A'_l = level in taxonomy of A'
A_l = level in taxonomy of A.

Of note in the above method is that the measure of similiarity is not defined at this stage in the research. Similarity can mean a number of things such as same attributes for semantic attributes based on some criteria.

V. Conclusion

The wide range of attacks available makes detection and defense a difficult prospect. Identifying an attack is the first step in combating it. By categorizing attacks into a self-organizing network, we are developing a quick method of attack identification. The application of fuzzy logic to selection of actionable rules creates a system that reasons dynamically about attack responses.

This initial work is being further refined and developed to include concepts from complex adaptive systems. We have built a small prototype that uses fuzzy logic to check classification of attacks against the taxonomy. Known attacks are being used to verify our approach. This allows further refinement of search and classification techniques. Once known attacks have been classified and our methods validated, we are moving to classify undocumented attacks as they are presented. With a working system that can be queried quickly, our eventual goals of real-time identification, self-organization of new taxonomies and classification of attack may be realized.

References

[1] Ducklin, Paul. The ABC of Computer Security. Retrieved April 12, 2003, from http://www.sophos.com/virusinfo/whitepapers/abc.html

[2] Symantec Corporation. Security Response. Retrieved March 15, 2003, from http:// securityresponse.symantec.com/
[3] SecurityFocus. What is BugTraq? Retrieved March 15, 2003, from http://www. securityfocus.com/popups/forums/bugtraq/intro.shtml
[4] NTBugTraq. NTBugTrack Home. Retrieved March 16, 2003, from http://ntbugtraq.ntadvice. com/
[5] SANS Institute. Computer Security Education and Information Security Training. Retrieved March 20, 2003, from http://www.sans.org/
[6] SecurityFocus. Vulns Archive. Retrieved March 12, 2003, from http://www.securityfocus.com/ bid
[7] National Institute of Standards and Technology. ICAT Metabase. Retrieved March 13, 2003, from http://icat.nist.gov/icat.cfm
[8] Taimur Aslam. A Taxonomy of Security Faults in the Unix Operating System. Master's Thesis, Purdue University, Department of Computer Sciences, August 1995
[9] M. Bishop. A taxonomy of unix system and network vulnerabilities. Technical Report CSE-9510, Department of Computer Science, University of California at Davis, May 1995.
[10] Shostack, Adam and Scott Blake. Towards a Taxonomy of Network Security Assessment Techniques, July 1999. Retrieved March 29, 2003, from htttp://razor.bindview.com/publish/ papers/taxonomy.html
[11] CERT. CERT® Advisory CA-2003-07 Remote Buffer Overflow in Sendmail. Retrieved April 2, 2003, from http://www.cert.org/advisories/ CA-2003-07.html
[12] Symantec Corporation. Backdoor.FTP_Ana.D. Retrieved April 13, 2003, from http://securityresponse.symantec.com/avcenter/venc/ data/backdoor.ftp_ana.d.html
[13] Symantec Corporation. Security Response. Retrieved April 21, 2003, from http://securityresponse.symantec.com/ avcenter/search.html
[14] SANS Institute, Knark: Linux Kernel Subversion. Retrieved April 24, 2003, from http://www.sans.org/resources/idfaq/knark.php
[15] Cole, Eric. *Hackers Beware*. New Riders Press,Indianapolis, IN, 2002.
[16] Berrueta, David Barruso. A Practical Approach for Defeating NMAP OS-Fingerprinting. Retrieved April 24, 2003, from http://voodoo. somoslopeor.com/papers/nmap.html
[17] Yen, John, Langari, Reza. *Fuzzy Logic, Intelligence, Control and Information*, Prentice Hall, 1999.