4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, November 17-19, 2005 (pp29-34)

A Fuzzy Taxonomic Approach for Classifying and Identifying System Attacks and Automating Attack Response

GREGORY VERT RENE DOURSAT

Department of Computer Science
University of Nevada, Reno
Reno, NV 89557, USA

Abstract: - Initial identification of attacks on computer systems is crucial to defending against them. A detailed classification system gives system administrators a tool for combating these attacks in the most effective fashion—by providing them with a specific path of action. There exists a tremendously wide range of attacks and defending against these requires an almost encyclopedic knowledge of their attributes and signatures. By relying on taxonomies that place entities in ever smaller and more precise groups, the user can rapidly identify common features and properties. However, different attacks can have similar attributes that can confuse classification. Therefore, we propose to use fuzzy logic both in the classification of attacks and an automated attack response system based on the selection of action rules.

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

1 Introduction

With the proliferation of viruses, worms, denial of service attacks and other vulnerabilities, network security experts have been continuously updating and reevaluating the methodologies used by malicious attackers. It has been estimated that about 10-20 new viruses appear daily [1]; due to this, security compromises are reported almost daily.

Several information resources are available that will notify users of new security holes on a subscription basis [2,3,4,5]. There are also several security databases that will let users browse the vulnerabilities for various software packages [6,7].

Companies such as Symantec, Security Focus and 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 viruses daily, these databases will soon become unwieldy.

2 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.

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 their classification has focused on a specific class of attacks. We will propose the classification of a broad range of computing attacks into a common hierarchy. This paper presents a novel new 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. This work has the potential to be highly beneficial to the security community.

3 Problem Solution

Attack databases, such as Security Focus' 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.

A taxonomy is a scientific technique to describe and organize categories of entities by representing objects in a hierarchy. 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 may be altered in the future as new attacks present themselves.

3.1 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

For attributes that have multiple values, we separate the values by commas, and define ranges using the "En dash" character (–). For example, a TCP port scan that targets ports 25 (SMTP), 80 (HTTP), and 1024 through 6000 would be defined as:

Network.Protocol.TCP.InPorts = 25, 80, 1024–6000

Our taxonomy includes 22 separate attack attributes. For the sake of brevity this list is not included in this paper. We will give a short overview of the taxonomy and structure in the following sections.

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

3.2 Node Actionable Response Rules

It is useful to have a series of responses that can potentially thwart an attack. Complex systems theory makes a general supposition that complex behaviors can be created through the composition and action of a series of smaller, simpler rules. For example ants obey simple rules, such as following a scent trail to a food source and dragging food back to the nest. In total, a number of ants peforming this task create the complex society we think of as an ant colony. 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 action rules 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

3.3 Attack Taxonomies

As stated in section 3.1, the first type of taxonomy developed here is based on common attributes of attacks that originated through a remote connection across a network.

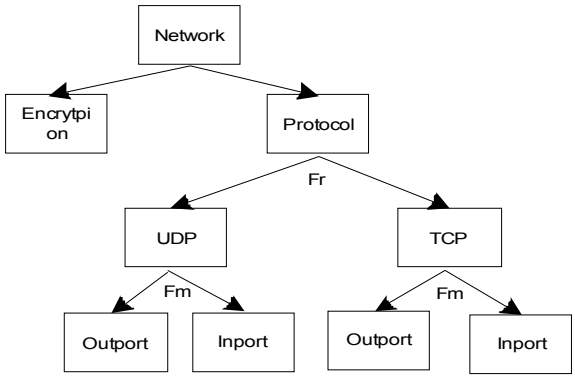


Fig. 1: Network Attribute Taxonomy

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.

Of note in the network taxonomy is that attacks originating outside of a network may depend on the target computer to be running a vulnerable daemon or service. This is represented in the tree by the TCP or UDP port number. If an attack requires a service or daemon, the attribute will reflect the default port number(s) of that service.

In our research 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.

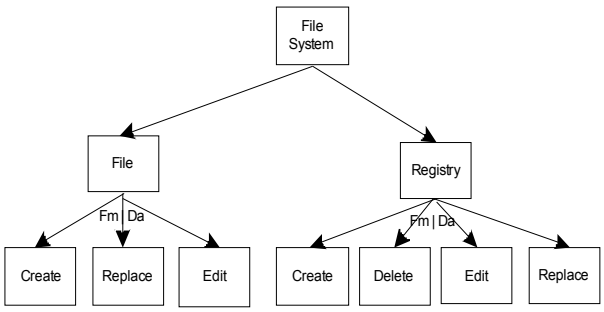


Fig. 2: File System Attribute Taxonomy

Another category of attacks are based on system exploits. Fig. 3 presents the *exploit attribute taxonomy* which was referenced in section 3.1. 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.

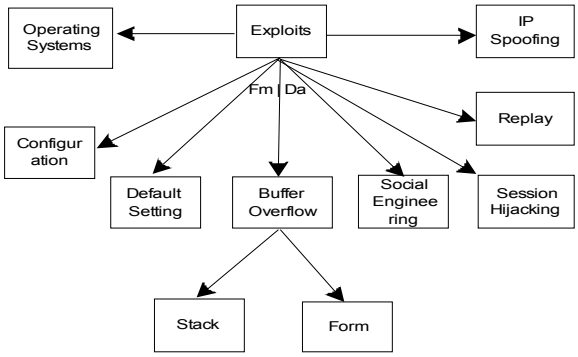


Fig. 3: Exploit Attribute Taxonomy

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.

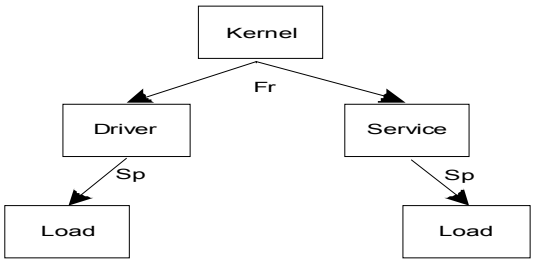
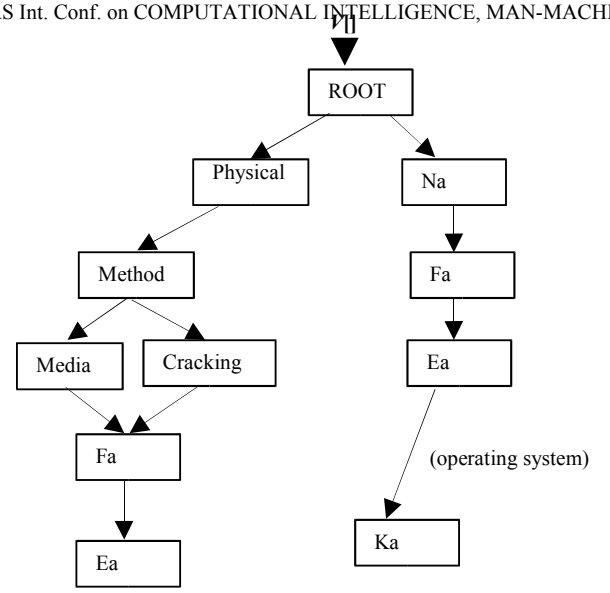


Fig. 4: Kernel Attribute Taxonomy

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.

Consolidating all of the above attacks into a single taxonomy produces what might be referred to as a taxonomic graph. This taxonomy is shown in Fig. 5, where each box represents the subtaxonomies presented in Fig. 1 through Fig. 4.



Ka – Kernel attribute tree
Ea – Exploit attribute tree
Na – Network attribute tree
Fa – File attribute tree

Fig. 5: Consolidated Taxonomic Graph

In Fig. 5 all leaf nodes are connected to the next subtaxonomic tree's root except for the right subtree connection from Ea to Ka. In this case, only the "operating system" node of the Exploit subtree is connected to the root of the Kernel subtree.

Input vector ***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. 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 ***V***[] of characterizing attributes, we utilize fuzzy linguistic variables to characterize the attack. As input data in ***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 the subtrees and tailored to a localized response to 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. 6 illustrates this point. In this case an attack can crisply belong to an Ea leaf node, or an Ka leaf node. However, with the application of the *FLV* variables, it is possible that an attack belongs to one or more leaf nodes.

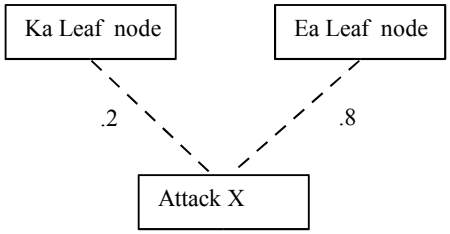


Fig. 6: Partial membership of an attack in multiple leaf nodes

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

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

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

Fig. 7: Algorithm to apply fuzzy linguistic variables

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].

5 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 an initial taxonomy, we are developing a quick method of 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. 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 goal of a real-time identification 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.