Fuzzy Intrusion Detection

John E. Dickerson, Jukka Juslin, Ourania Koukousoula, Julie A. Dickerson
Electrical and Computer Engineering Department
Iowa State University
Ames, IA, USA
jedicker,juslin,koukouso,julied@iastate.edu

Abstract

The Fuzzy Intrusion Recognition Engine (FIRE) is a network intrusion detection system that uses fuzzy systems to assess malicious activity against computer networks. The system uses an agent-based approach to separate monitoring tasks. Individual agents perform their own fuzzification of input data sources. All agents communicate with a fuzzy evaluation engine that combines the results of individual agents using fuzzy rules to produce alerts that are true to a degree. Several intrusion scenarios are presented along with the fuzzy systems for detecting the intrusions. The fuzzy systems are tested using data obtained from networks under simulated attacks. The results show that fuzzy systems can easily identify port scanning and denial of service attacks. The system can be effective at detecting some types of backdoor and Trojan horse attacks.

1. Introduction

Network intrusion detection (NID) is the process of identifying network activity that can lead to the compromise of a security policy. Most commercial NID systems use a form of intrusion detection called "misuse detection" that compares data in the network stream against a database of known attack signatures. These systems are usually only effective when prior knowledge of the detailed characteristics about various intrusion techniques is available. We would prefer to be able to identify potentially malicious activity without prior knowledge of what form the attacks will take.

Anomaly detection attempts to spot malicious activity by looking for unusual events in the data being monitored. The difficulty in anomaly detection is knowing what features in the input to monitor. Some features may be irrelevant to certain intrusion detection scenarios. Some types of attacks are difficult to identify unless inputs from multiple monitors are combined. The next generation of intrusion detection tools will need to be able to perform correlation analysis of multiple inputs.

This research explores using fuzzy systems as the correlation engine for an intrusion detection system. Fuzzy systems have several important characteristics that suit intrusion detection very well.

- Fuzzy systems can readily combine inputs from widely varying sources.
- Many types of intrusions are cannot be crisply defined (e.g. at what threshold should an alarm be set?)
- The degree of alert that can occur with intrusions is often fuzzy.

This paper presents the design of the FIRE system and discusses how the fuzzy agents are used to perform correlation of multiple inputs for intrusion detection.

2. Architecture

FIRE utilizes the Autonomous Agents for Intrusion Detection (AAFID) architecture developed at Purdue by Zamboni, et al. AAFIDS implements a framework for the architecture of a distributed Intrusion Detection System (IDS) using independent entities working collectively, and it is developed by Purdue University [1]. This system provides useful characteristics like configurability, scalability, and flexibility. There are three distinct independent components of the architecture called agents, transceivers, and monitors. They each play different roles as part of the system, but their function can be altered only by the operating system not another process; thus they are autonomous.

An agent monitors processes of a host and reports abnormal behavior to a transceiver. It can communicate with another local agent only through a transceiver. A transceiver controls local agents and acts as the external communication tool between a host and a monitor. A transceiver can perform appropriate data processing (analysis and reduction), and report to monitors or other agents. A monitor is similar to a transceiver but it also controls entities (agents, transceivers, and lower level monitors) in several hosts. Monitors combine higher-level reports, correlate data, and send alarms or reports to the User Interface.

Several agents were developed for FIRE:

TCPconn Agent – Monitors TCP connections between hosts on the network looking for unusual connection patterns.

UDPconn Agent – Looks for unusual traffic involving UDP data.

ICMPconn Agent – Monitors ICMP traffic.

PortAgent – Monitors which ports are in service on the network and watches for unusual services and service/host combinations.

Additionally, a monitor was developed to combine the outputs for detecting attacks. The monitor determines the fuzzy threats present by applying fuzzy rules to the inputs obtained from the FIRE agents.

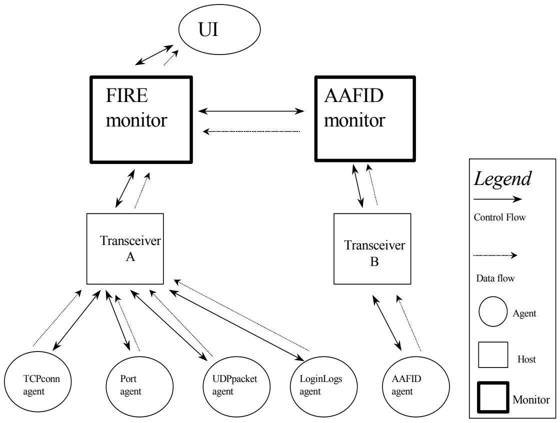

Figure 1. FIRE architecture. A FIRE system includes one or more specialized agents and a fuzzy threat monitor. The system utilizes the AAFIDS architecture.

Data Collection

Choosing the best data elements to monitor in the network stream is critical to the effectiveness of the intrusion detection system. Since FIRE is intended to on the network packet header data rather than the contents of network packets. FIRE concentrates on the three main internet protocols: TCP, UDP, and ICMP. The system monitors all traffic during a two week sliding collection window. For the most part, web traffic is ignored for this scope of this investigation.

Data reduction is critical when monitoring network data over a lengthy period. In order to conserve storage space, FIRE records only information about the type of connection, the source and destination, the services used, length of the connection (in time and in number of packets), and connection completion status. The length of the collection period allows the system to monitor typical patterns of behavior in the network traffic over time.

The data is logged on a data collection host where one or more FIRE agents are present. All FIRE agents that perform network monitoring utilize the same network data logs as input, though each agent may be monitoring different aspects of the log data.

The individual FIRE agents each monitor a specific type of data to create a set of observed metrics about its data source. In general, these metrics are reduced to quantitative values observed in a discrete time interval. We use an observation time interval of 10 minutes, though this time is chosen somewhat arbitrarily. For instance, one of the metrics observed by the TCPconn agent is the total number of connections observed in the 10 minute observation window. All together, the five FIRE agents monitor a total of 64 separate metrics. Over time, the metrics are used to establish the values for the fuzzy sets in the intrusion recognition engine.

3. Analysis

Fuzzy Sets

Once enough data is collected over a two-week collection period, fuzzy inputs sets from each metric are produced. In general, the extents and midpoints of the membership functions were determined with a fuzzy C-means algorithm. Though, as we shall see, some metrics produce sparse variation that may require more simple statistical models to define the sets. There are five membership functions in each input set: LOW, MED-LOW, MEDIUM, MED-HIGH, and HIGH.

The output membership functions are uniformly distributed in the range from 0.0 to 1.0. Figure 2 shows a typical set of output membership functions for an alert.

Fuzzy Rules

With the fuzzy input sets defined, the next step is to write the rules for detecting each type of attack. A collection of fuzzy rules with the same input and output variables is called a fuzzy system. We assume that the security administrator can use their expert knowledge to help create a set of rules for each attack.

The rules are created using the fuzzy system editor contained in the *Matlab* Fuzzy Toolbox. This tool contains a graphical user interface that allows the rule designer to create the member functions for each input or output variable, create the inference relationships between the various member functions, and to examine the control surface for the resulting fuzzy system. But it is not expected that the rule designer will rely solely on intuition to create the rules. Visual data mining can help the rule designer understand data features are most relevant to detecting different kinds of attacks.

Fuzzy C-Means Clustering

The fuzzy *c*-means (FCM) algorithm minimizes the objective function [2].

$$J(\mathbf{U},\mathbf{V}) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m \left\| x_k - v_i \right\|^2$$

$$\text{subject to } u_{ij} \in [0,1] \text{ and } \sum_{i=1}^{c} u_{ik} = 1 \ \forall \ k \tag{1}$$

U is the partition matrix that shows to what degree the *k*-th data point x_k belongs to each cluster as measured by its distance from the prototype of the *i*-th cluster, v_i. m is a weighting exponent. c is the number of clusters and n is the number of data points. This algorithm is given in [2].

4. Results

Several intrusion scenarios were used to test FIRE. The three that will be discussed here are:

1. Host and port scanning
2. Denial of service
3. Unauthorized servers

Host and Port Scanning

Attackers frequently conduct host and port scans as precursors to other attacks. An intruder will attempt to determine the existence of hosts on a network or whether a particular service is in use. A host scan is usually characterized by a unusual number of connections to hosts on the network from an uncommon origin. The scans may utilize a variety of protocols. We use an identifier called an SDP to represent a unique connection between a source, destination, and a service port. A sample rule from a fuzzy system for detecting a "stealthy" scan to identify the existence of servers using port N on a network can be written as:

If the COUNT of UNUSUAL SDPs on Port N is HIGH
And the COUNT of DESTINATION HOSTS is HIGH
And the COUNT of SERVICE Ports observed is MEDIUM-LOW
Then Service Scan of Port N is HIGH

A complete fuzzy system for this rule set is shown in Figure 3.

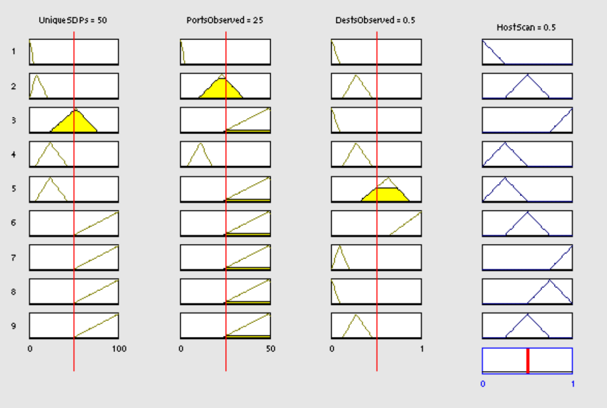

Figure 2. Fuzzy system for detecting a host scan. The inputs to the system are the number of destination hosts observed, the number of service ports contacted, and the number or soure/dest/port combinations observed.

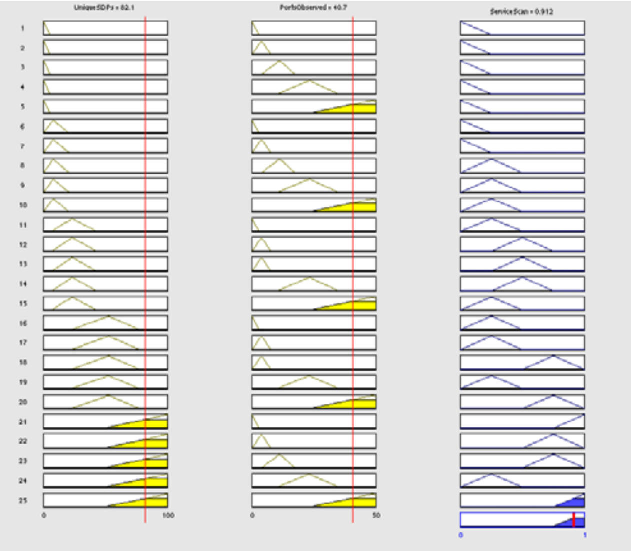

Figure 3. Fuzzy system for detecting a port scan. The inputs to the system are the number of unique source/dest/port combinations and the number of service ports observed, and the number of destination hosts.

Each fuzzy system is assumed to have five member functions, each with a triangular distribution. For simplicity, we assume that the extents of each rule lie at the center point of the adjacent rule. This assures complete coverage in the input domain. The input domain is clipped at the minimum of the leftmost and maximum of the rightmost rules so extremely high or low values will still lie within the rule domain.

The next step is to enumerate the fuzzy sets for each type of scan. The fuzzy c-means algorithm was applied to the data gathered during the two-week observation period prior to running the attacks. This established the values used to quantify the rules in figures 2 and 3. Representative resulting membership functions for number of service ports observed, and number of unique SDPs are shown in Figures 4 and 5 respectively.

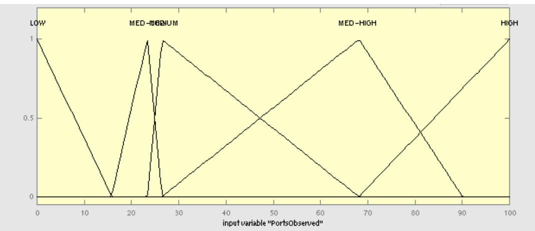

Figure 4. Membership functions for the number of observed service ports obtained using fuzzy C-means algorithm.

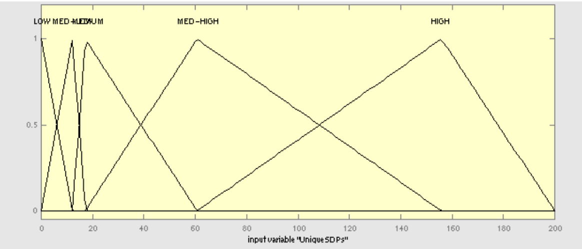

Figure 5. Membership functions for the number of observed SDPs obtained using fuzzy C-means algorithm.

Denial of Service Detection

A common attack scenario consists of an attacker overwhelming a target machine with too much data. This chokes the target and prevents it from performing its intended role. Denial of service (DoS) attacks can take a variety of forms and use many different protocols [3]. We developed a representative fuzzy system for a common DoS attack based on ICMP traffic congestion. To test the system, we launched an ICMP DoS attack called *pingflood* against a target in a controlled environment, collected the network traces and input the resulting data to the fuzzy system. A representative rule used in the ICMP DoS system can be written as:

> IF COUNT of UNUSUAL SDTs is HIGH
> AND COUNT of ICMP ERQs is HIGH
> THEN pingflood ALERT is HIGH

Where SDTs are the combined [Source, Destination, ICMP Type] identifier of ICMP packets. The complete fuzzy system for detecting this type of attack is shown in Figure 6.

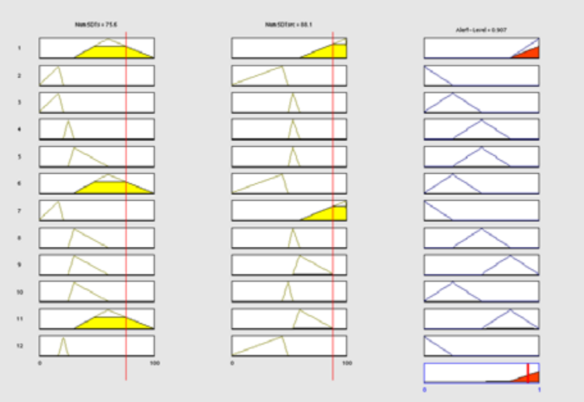

Figure 6. Fuzzy system for ICMP denial of service network attack. The inputs are the number of unique SDTs observed, and the number of hosts observed.

The fuzzy system above was able to sense a *pingflood* attack from network traffic data. Because the ICMP traffic of this nature is generally uncommon on networks unless an ICMP flood attack is occurring, the likelihood of false positives is low.

Unauthorized Servers Detection

Another intrusion detection scenario that is potentially more damaging than the prior two scenarios is the situation where an attacker has managed to invade a system and install a backdoor or Trojan horse program that can lead to further compromise. Telltale activity that would help identify such intrusions would include identifying unusual service ports in use on the network, unusual numbers of connections from foreign or unfamiliar hosts, and/or unusual amounts of network traffic load to/from a host on the network. With FIRE, we identify servers as the combination of host IP address and specific service port. A representative rule for this detection scenario is:

IF COUNT of UNIQUE SERVERS is MEDIUM HIGH
And UNIQUENESS of SERVERS is HIGH
And UNIQUE FOREIGN SDPs is MEDIUM HIGH
THEN Unauthorized Server ALERT is HIGH

The control surface for this system is shown in Figure 7. It is important to note that in practice the number of unique servers observed on a network will usually be zero for a large part of the two week observation period. Any number of new servers greater than zero is cause for investigation, if not alarm. However, simply using a fuzzy C-means algorithm to quantify the membership functions will not reflect this discontinuity adequately. Care must be taken to design the fuzzy system so that system response adequately reflects the alert level.

To test this fuzzy system, we gathered data for two weeks then installed a backdoor SSH server on an uncommon port (789), then connected to the backdoor several times and transferred some files to the machine over that port. The port monitoring agent correctly detected that one unique (previously unobserved) server existed on the network. However, when the fuzzy C-means algorithm alone was used to quantify the membership functions for the number of unique servers, the output alert level was not high enough to trigger concern. Approximately 8 percent of the observation periods during the two-week collection period issued a similar threat output, suggesting that the false negative rate for this system was too high.

However, when the fuzzy system was modified by hand so that any positive number of unique servers would lead to at least a medium high alert level, the percent of observation periods that would have issued a similar response as the simulated attack dropped to approximately 1.5 percent. This suggests that an unsupervised fuzzy rule design will respond with a considerably higher false positive rate if the input threat includes non-linear alert responses.

5. Conclusions

Anomaly-based network intrusion detection is a complex process. The variety in the network data stream, the amount of data to be processed, and the subtle and ever-changing ways that intruders penetrate systems all conspire to complicate the task. Nonetheless, this research has shown that there are several broad classes of intrusion behavior that can be detected with a general anomaly-based approach using fuzzy systems. The metrics monitored by the fuzzy agents at this phase in the system development are clearly sufficient for detecting many types of scanning activity and denial of service attacks. More specific intrusions such as backdoor installations are also possible though they may require more expert knowledge to design an accurate fuzzy detection rule set [4]. More specialized agents and detailed metrics will be necessary to identify more subtle attacks. While this research does not solve the problem of finding all network-based invasions, the generality of fuzzy intrusion detection holds promise as a high-level intrusion detection scheme that works in conjunction with detailed misuse detection systems. Most importantly, this research has laid a solid groundwork for fuzzy intrusion detection and revealed promising areas of continued exploration.

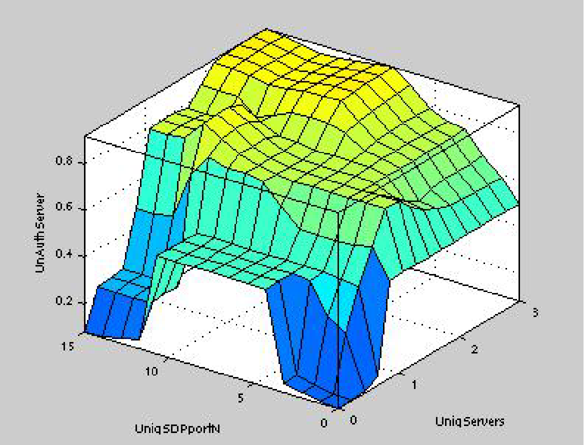

Figure 7. Control surface for fuzzy system to detect unauthorized servers. The front axis is the number of unique source/dest addresses observed for port N, the retreating axis is the number of unique servers observed in use during the observation period.

6. References

[1] Zamboni, D. et al, "An Architecture for Intrusion Detection using Autonomous Agents," *COAST Technical Report 98/05*, COAST Laboratory. Purdue University. June 11, 1998.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, New York, 1981.

[3] S. Northcutt, *Intrusion Signatures and Analysis*, New Riders, Indianapolis, Indiana, 2001, pp 189-211.