

# CeNISA

ISRCS 2009

## Computationally efficient Neural Network based Intrusion Security Awareness

D. Todd Vollmer and Milos Manic



University of Idaho  
Idaho Falls

# Problem

How to present the results of an anomaly based system to an operator by making use of information from a rule based intrusion detection system and improve execution performance.

# Intrusion Detection Systems

- Network or Host based
- IDS vs. IPS
- Two types of network intrusion detection
  - Anomaly (behavior)
  - Rule (knowledge)

# Anomaly IDS Pro/Con

- Can detect new or variants of attacks
- Less dependent on O.S. Specific characteristics
- Can have high false positives
- Behavior may change over time
- Example anomaly data may be difficult to find

# Rule IDS Pro/Con

- Specific attacks and vulnerabilities.
- Potentially low false positive rates.
- Detailed rules allow for more directed action.
- Expensive rule maintenance.
- Gathering new attack information after fact.
- Variations on rules are not caught.

# SNORT®

- An open source IDS created by Martin Roesch
- Rule based
  - Protocol matching
  - Content searching

Snort is a registered trademark of Sourcefire, Inc.

# Snort Rules

- `<action><protocol><SIP><Sport><direction><DIP><Dport>`  
(`<rule options>`)
- Alert tcp any any → 192.168.1.0/24 any (content:"ELHO")
- Rule Sets
  - Sourcefire Vulnerability Research Team (VRT)
  - Community Rules
  - Emerging threats
- Many Options (important keyword classtype)

# Classifications

<i>Class type</i>	<i>Class type</i>
attempted-admin	rpc-portmap-decode
attempted-user	successful-dos
kickass-porn	successful-recond-largescale
policy-violation	successful-recon-limited
shellcode-detect	suspicious-filename-detect
successful-admin	suspicious-login
successful-user	system-call-detect
trojan-activity	unusual-client-port-connection
unsuccessful-user	web-application-activity
web-application-attack	icmp-event
attempted-DOS	misc-activity
attempted-recon	network-scan
bad-unknown	not-suspicious
default-login-attempt	protocol-command-decode
denial-of-service	string-detect
misc-attack	unknown
non-standard-protocol	tcp-connection

# CeNISA Algorithm

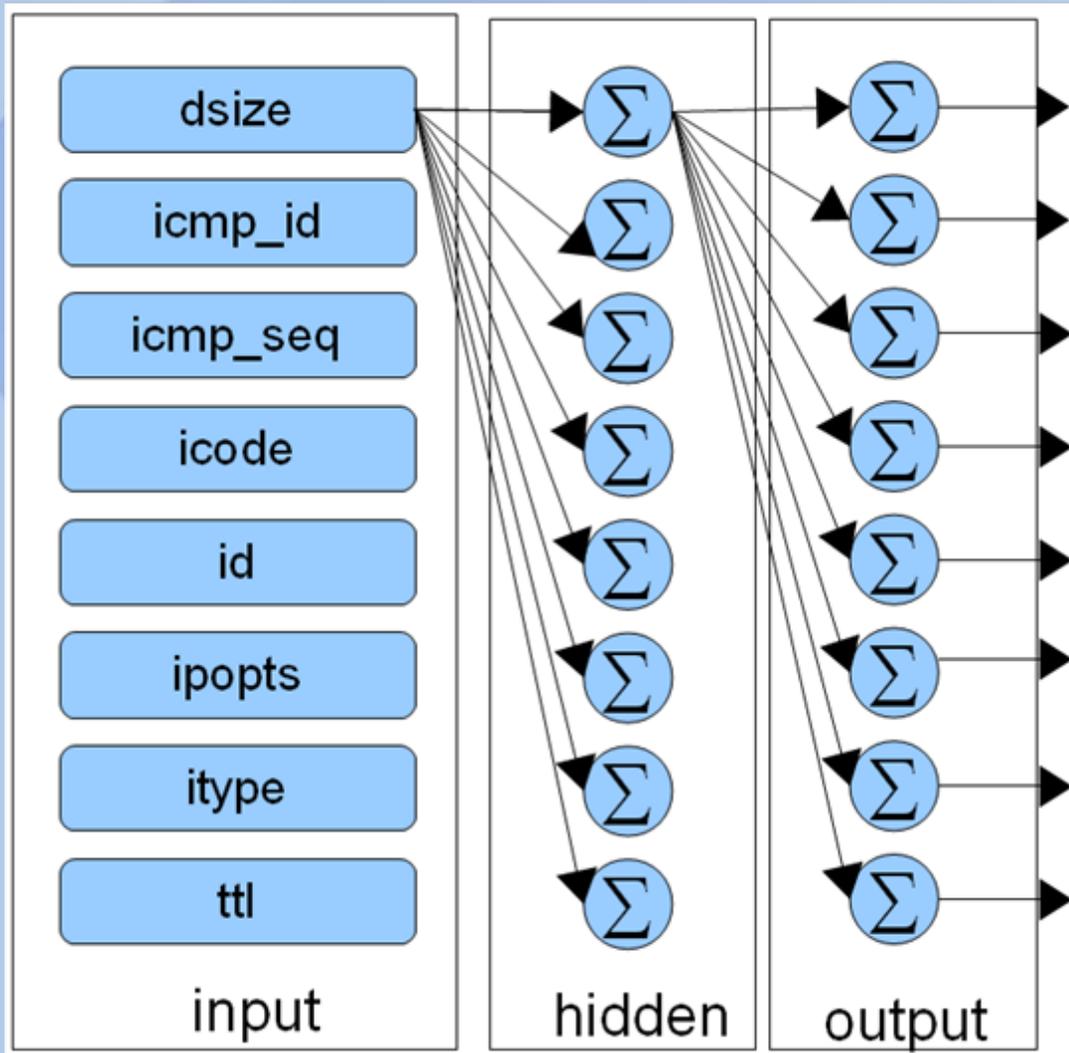
The general process for creating an EBP network is described as follows:

1. Define a feature vector, gather the training data and present it to the network as input.
2. Determine the number of hidden and output layers.
3. Using an input feature vector compare the network's output to the desired output. Calculate the error from each output node.
4. Incrementally adjust the weights of each node using the error calculations as a basis of calculation.
5. Using the error calculations for each node, feed the values back through the layers adjusting weights accordingly.

Repeat steps 3 - 5 until some acceptable error level is reached.

# CeNISA Algorithm

## Neural Network



Classification

# Testing Set

- ICMP network data proof of concept
  - Rule set contained rules for 8 class types. (3 single entries)
  - 5 test cases created to exactly match 5 different class types. 100% identification against these tests.
  - Removed matching 5 rules.

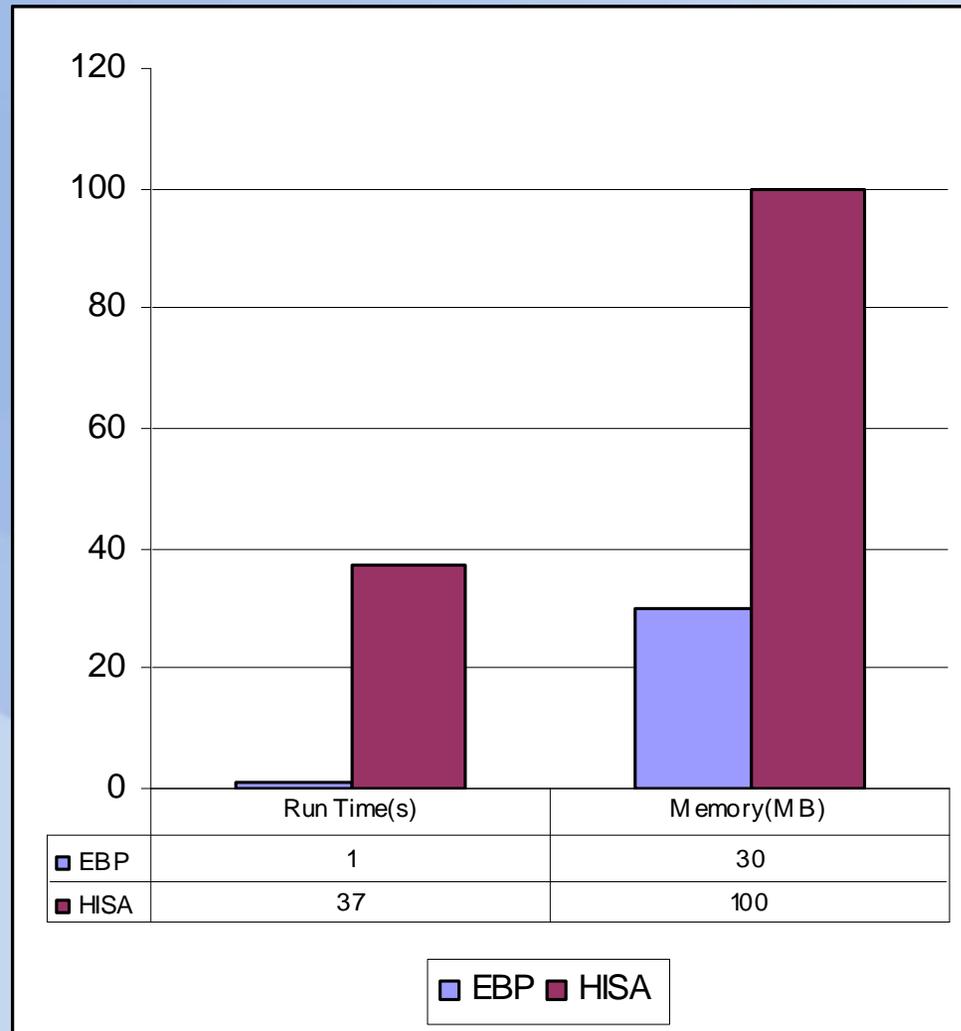
# ICMP Class Types

- Attempted-dos
- Attempted-recon
- Bad-unknown
- Trojan-activity
- misc-activity

# Results

<i>Correct ClassType</i>	<i>Identified ClassType</i>	<i>% Match</i>
attempted-dos	attempted-dos(3)	100%
attempted-recon	Attempted-recon(4) network-scan(1)	80%
bad-unknown	bad-unknown(2) attempted-recon(3) misc-activity (28)	6%
trojan-activity	attempted-user(1)	0%
misc-activity	misc-activity(1)	100%

# Performance Comparison



# Conclusions

- 60% identification rate on test data matches HISA Algorithm. EBP had a 75% identification rate in training/testing cycle.
- Rule definition is important, many misc-activity rules (71% ICMP).
- Computationally efficient in time and memory compared to previous solution.