DATA MINING FOR INTRUSION DETECTION

Aleksandar Lazarević,
Jaideep Srivastava, Vipin Kumar

Army High Performance Computing Research Center
Department of Computer Science
University of Minnesota

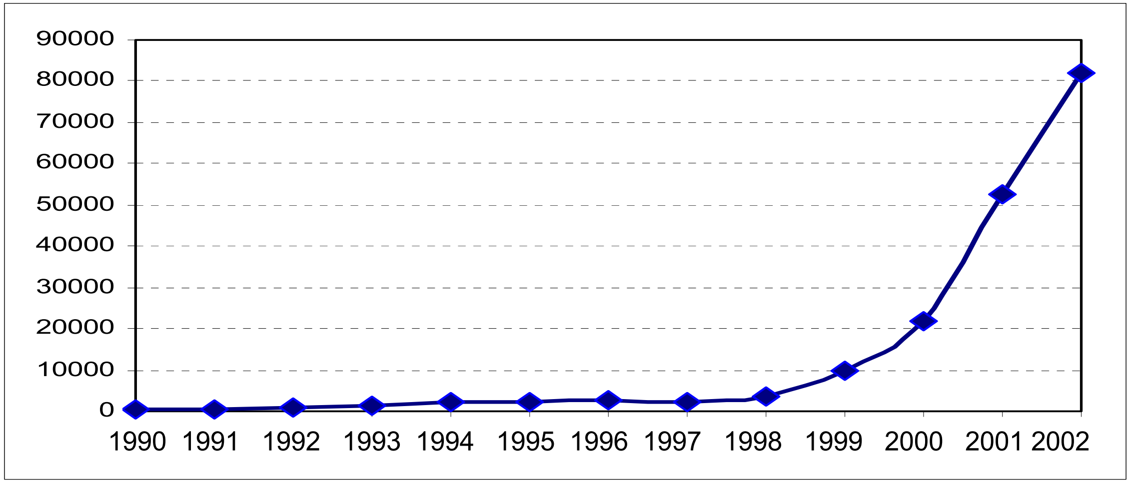
Tutorial on the Pacific-Asia Conference on Knowledge Discovery in Databases 2003

AHPCRC

Introduction

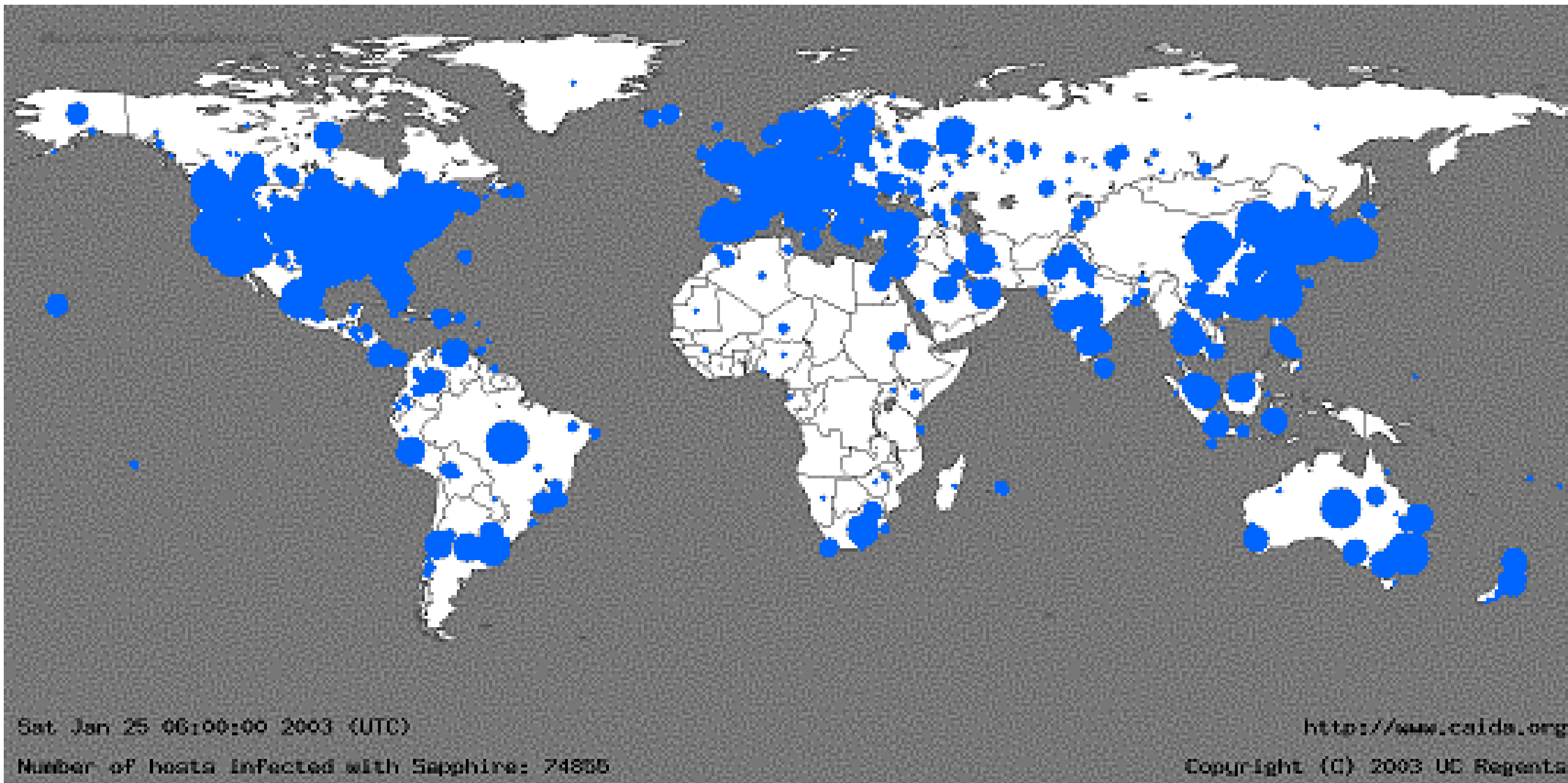
- Due to the proliferation of high-speed Internet access, more and more organizations are becoming vulnerable to potential cyber attacks, such as network intrusions
- Sophistication of cyber attacks as well as their severity has also increased recently (e.g., Code-Red I & II, Nimda, and more recently the SQL slammer worm on Jan. 25)

Incidents Reported to Computer Emergency Response Team/Coordination Center (CERT/CC)



The Spread of the Sapphire/Slammer Worm

- **The geographic spread of Sapphire/Slammer Worm 30 minutes after release**

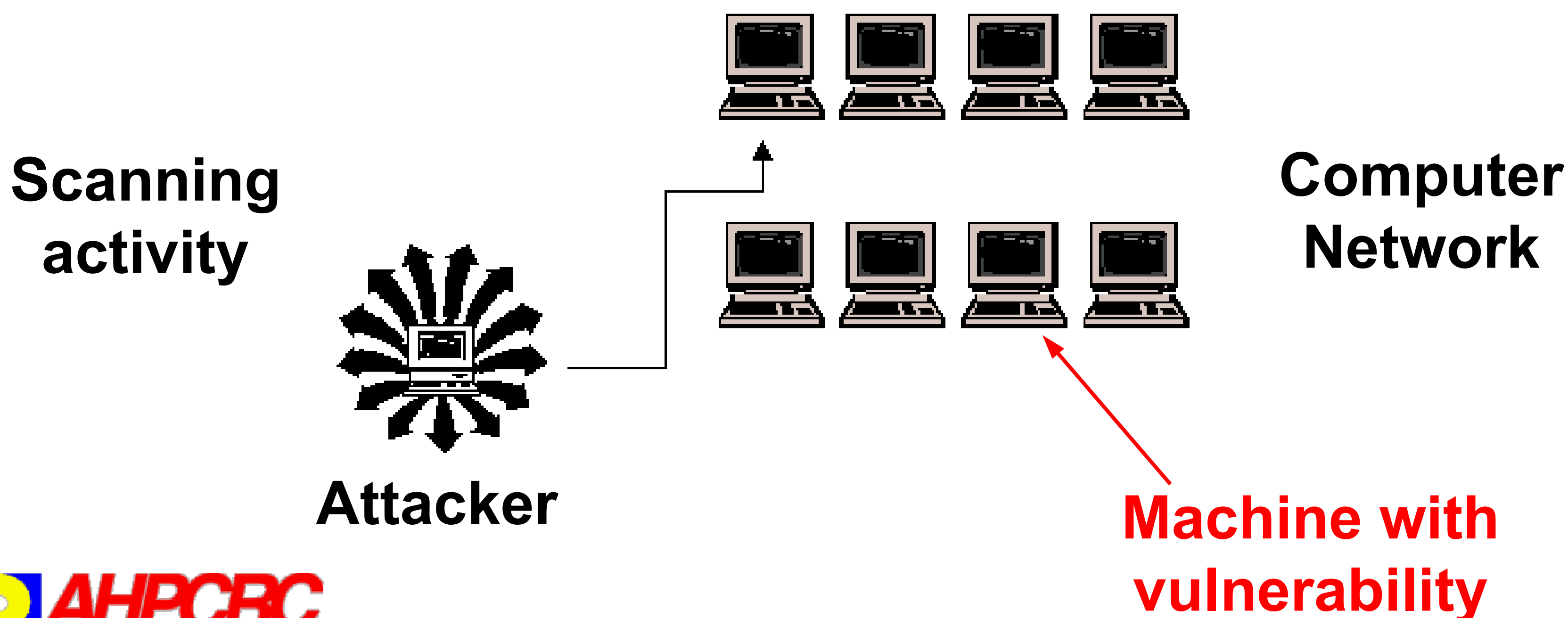


AHPCRC

Source: www.caida.org

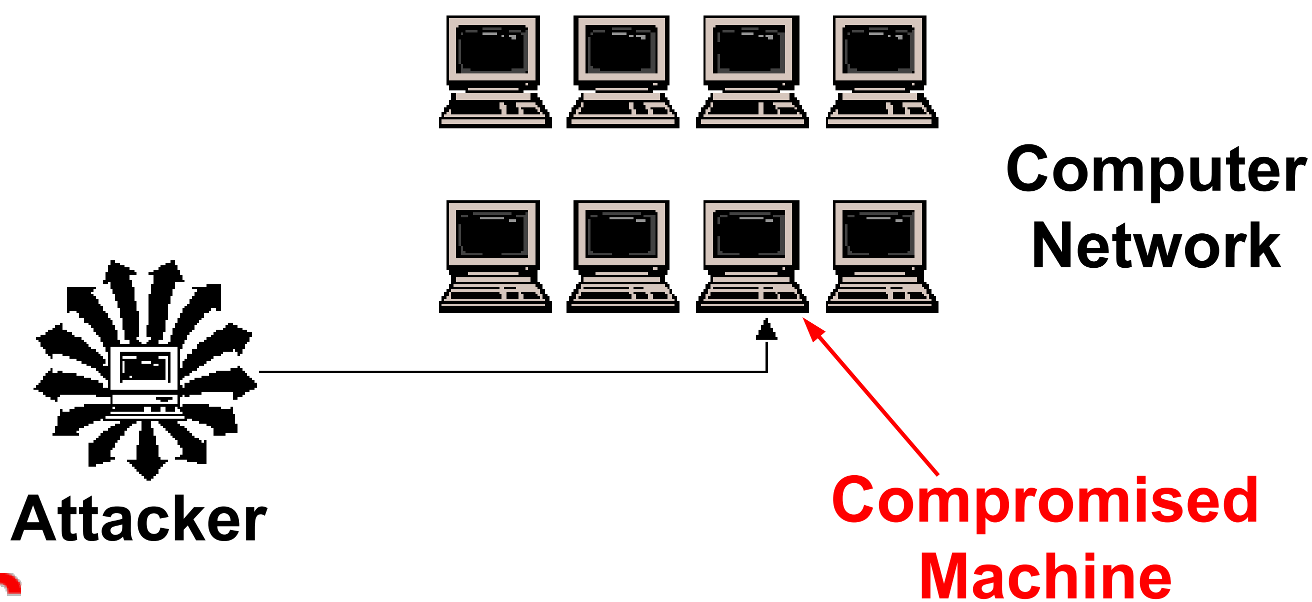
Cyber Attacks - Intrusions

- Cyber attacks (intrusions) are actions that attempt to bypass security mechanisms of computer systems. They are caused by:
  - Attackers accessing the system from Internet
  - Insider attackers - authorized users attempting to gain and misuse non-authorized privileges
- Typical intrusion scenario



Cyber Attacks - Intrusions

- Cyber attacks (intrusions) are actions that attempt to bypass security mechanisms of computer systems. They are caused by:
  - Attackers accessing the system from Internet
  - Insider attackers - authorized users attempting to gain and misuse non-authorized privileges
- Typical intrusion scenario



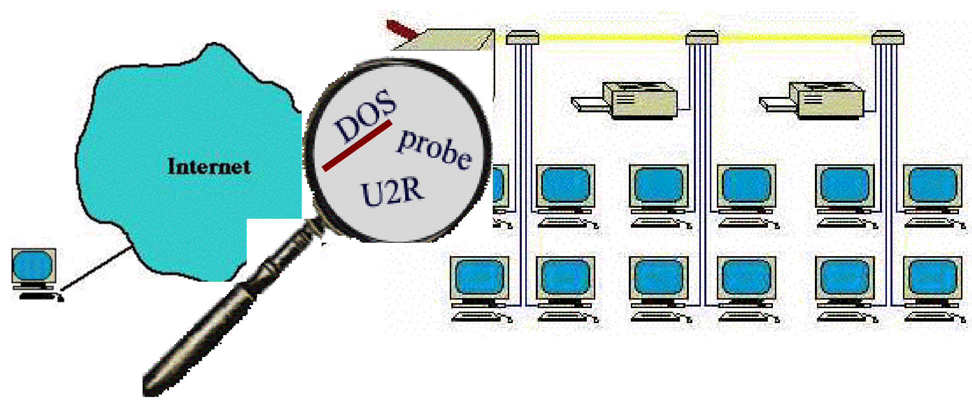
AHPCRC

Why We Need Intrusion Detection?

- Security mechanisms always have inevitable vulnerabilities
- Current firewalls are not sufficient to ensure security in computer networks
  - "Security holes" caused by allowances made to users/programmers/administrators
  - Insider attacks
  - Multiple levels of data confidentiality in commercial and government organizations needs multi-layer protection in firewalls

Intrusion Detection

- **Intrusion Detection:** Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to bypass the security mechanisms of a computer or network ("compromise the confidentiality, integrity, availability of information resources")
- **Intrusion Detection System (IDS)**
  - combination of software and hardware that attempts to perform intrusion detection
  - raise the alarm when possible intrusion happens



Traditional Intrusion Detection Systems

- **Traditional intrusion detection system (IDS) tools (e.g. SNORT) are based on signatures of known attacks**
  - **Example of SNORT rule (MS-SQL “Slammer” worm)**

```
any -> udp port 1434 (content:"|81 F1 03 01 04 9B 81 F1 01|";
content:"sock"; content:"send")
```

www.snort.org

- **Limitations**
  - **Signature database has to be manually revised for each new type of discovered intrusion**
  - **They cannot detect emerging cyber threats**
  - **Substantial latency in deployment of newly created signatures**
- **Data mining based IDSs can alleviate these limitations**

AHPCRC

Taxonomy of Computer Attacks

- **Intrusions can be classified according to several categories:**
  - ***Attack type*** **(Denial of Service (DoS), Scan, worms/trojan horses, compromises (R2L, U2R), …)**
  - ***Number of network connections*** **involved in the attack**
    - **single connection cyber attacks**
    - **multiple connections cyber attacks**
  - ***Source of the attack***
    - **multiple vs. single**
    - **inside vs. outside**
  - ***Environment*** **(network, host, P2P, wireless networks, …)**
  - ***Automation*** **(manual, automated, semi-automated attacks)**

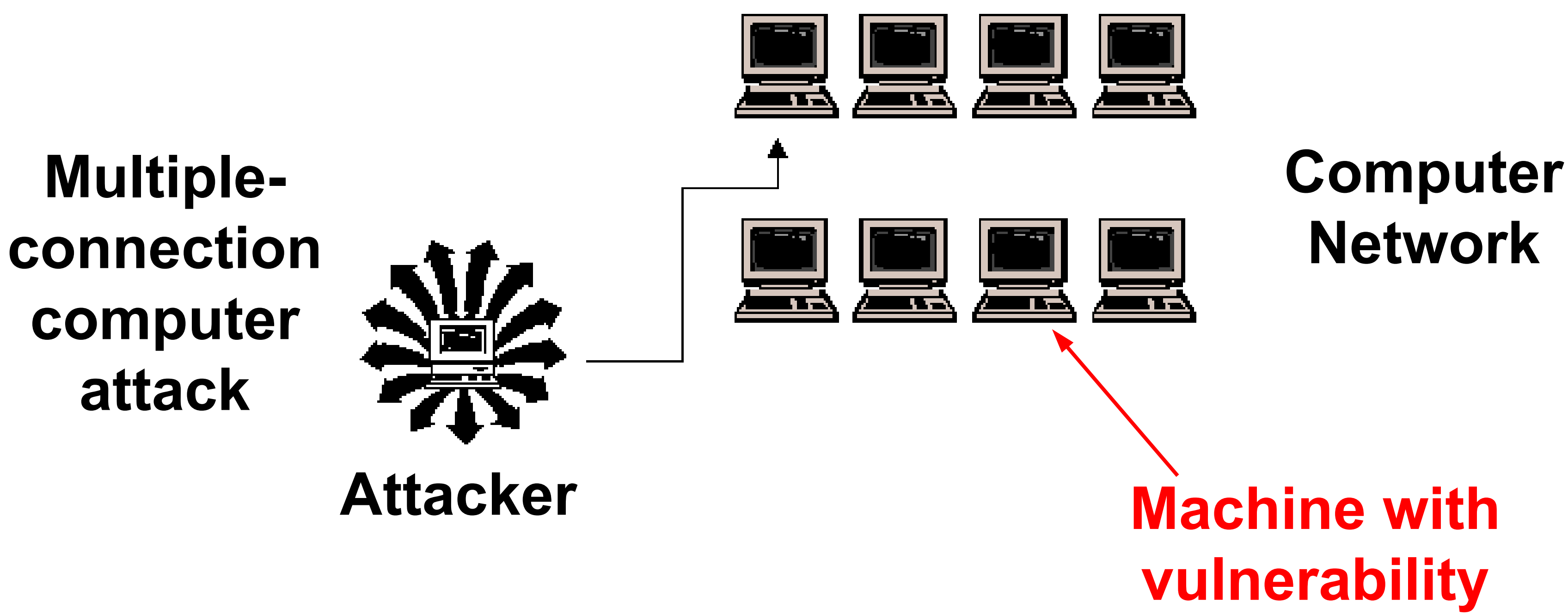
Types of Computer Attacks

- **DoS (Denial of Service) attacks**
  - DoS attacks attempt to shut down a network, computer, or process, or otherwise deny the use of resources or services to the authorized users
  - Distributed DoS attacks
- **Probe (probing, scanning) attacks**
  - Attacker uses network services to collect information about a host (e.g. list of valid IP addresses, what services it offers, what is the operating system)
- **Compromises** - attackers use known vulnerabilities such as buffer overflows and weak security to gain privileged access to hosts
  - **R2L (Remote to Login)** attacks - attacker who has the ability to send packets to a machine over a network (but does not have an account on that machine), gains access (either as a user or as a root) to the machine and does harmful operations
  - **U2R (User to Root)** attacks - attacker who has access to a local account on a computer system is able to elevate his or her privileges by exploiting a bug in the operating system or a program that is installed on the system
- **Trojan horses / worms** – attacks that are aggressively replicating on other hosts (worms – self-replicating; Trojan horses are downloaded by users)

AHPCRC

Number of connections involved in attacks

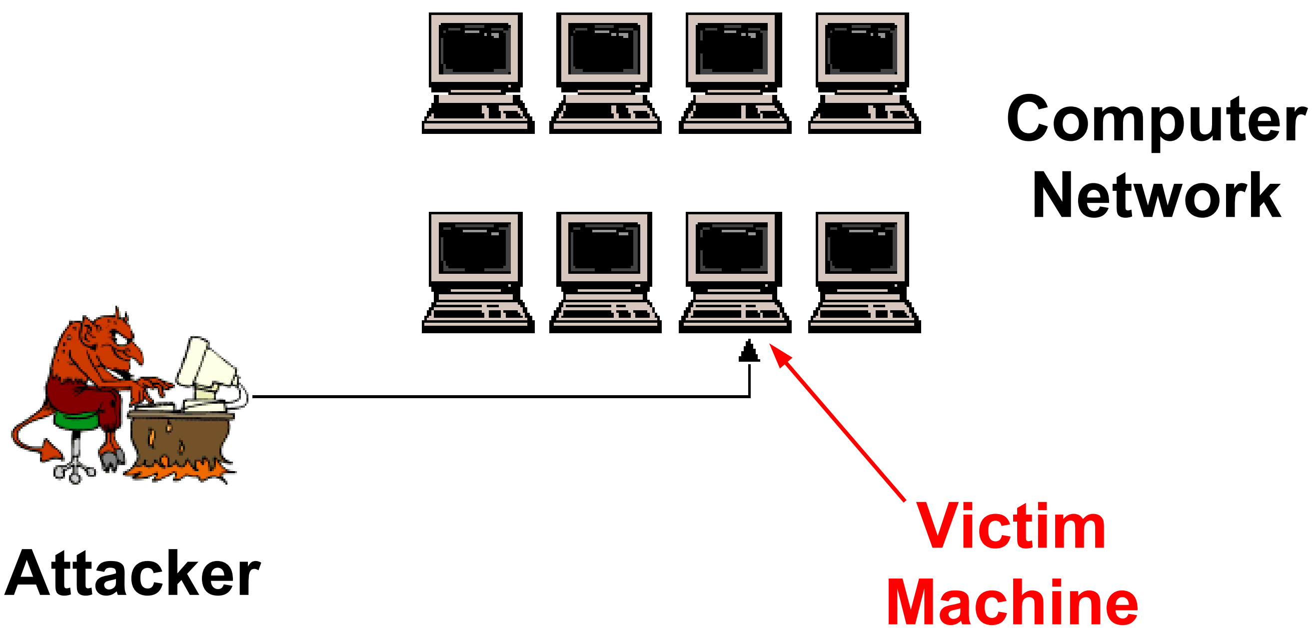
- Generally two types of cyber attacks in the computer networks:
  - attacks that involve multiple network connections (bursts of connections)
  - attacks that involve single network connections



AHPCRC

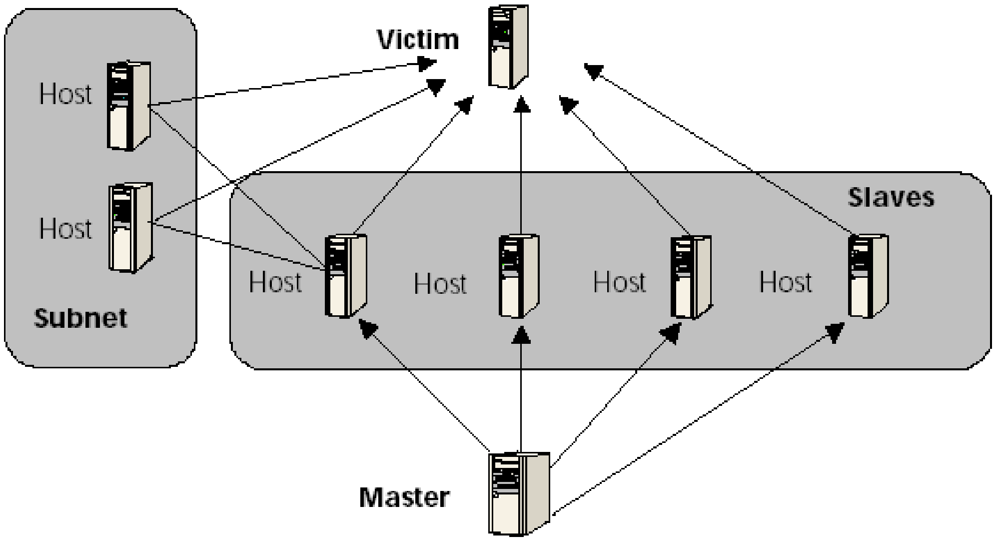
Number of connections involved in attacks

- **Single connection attack**



Source of Computer Attacks

- Attacks may be launched from single location or from several different locations
- Attacks may be also targeted to single or many different destinations
- Need to analyze network data from several sites in order to detect these distributed attacks.
  - Single source attacks
  - Distributed/Coordinated attacks



Environment of Computer Attacks

- Attacks may be categorized according to the environment where they occur
  - Network intrusions (intrusions in computer networks)
  - Intrusions on the host machine (single computers)
  - Intrusions in P2P environment
    - connected computers act as peers on the Internet, nothing else than clients
    - they are cut off from the DNS system since they do not have fixed IP address, and therefore difficult to trace the attack source
  - Intrusions in wireless networks
    - Physical layer is less secure than in fixed computer networks
    - Mobile nodes do not have fixed infrastructure
    - There are no traffic concentration points where packets can be monitored
  - …

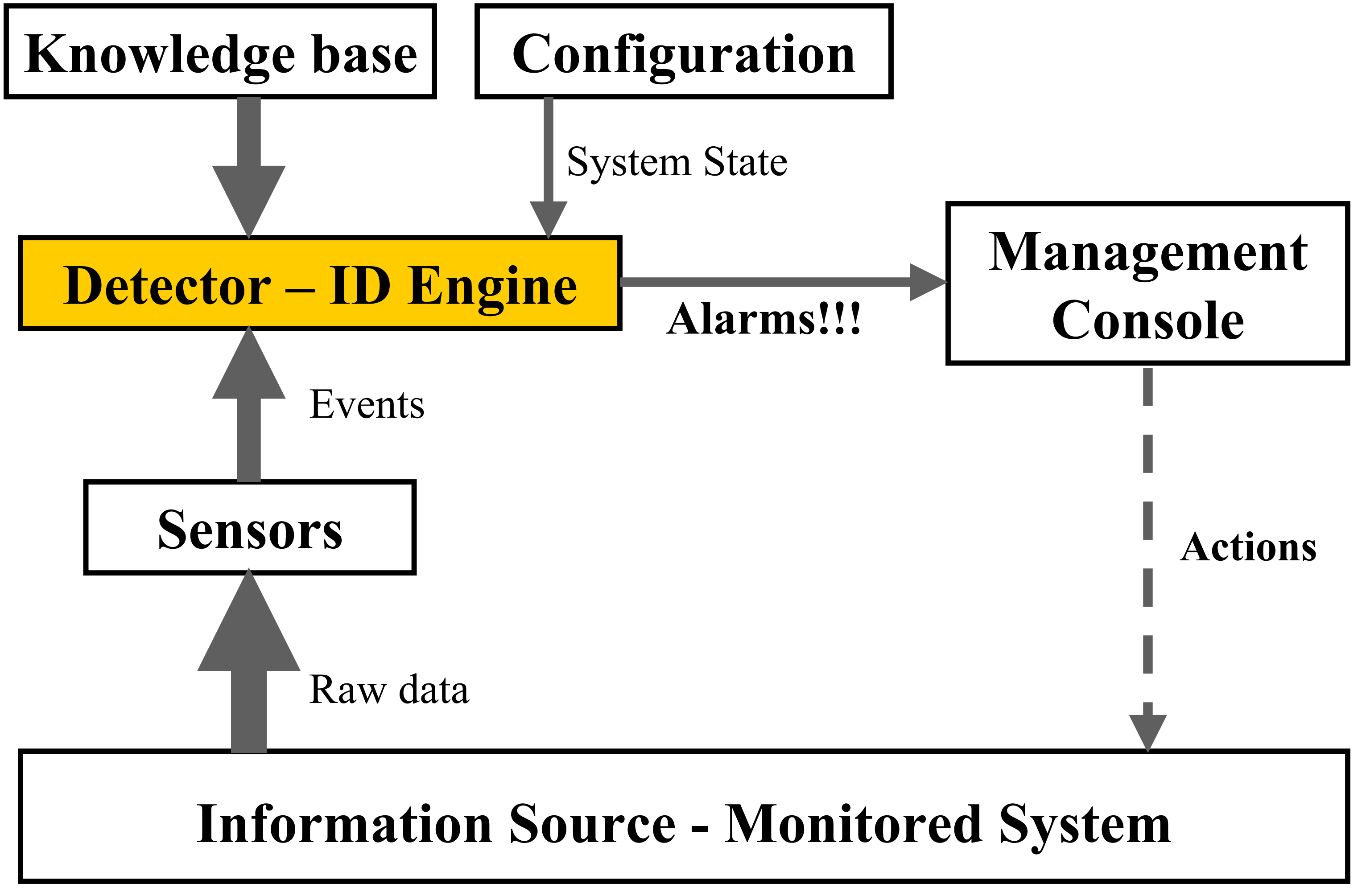
Automation of Computer Attacks

- **Wide-spread availability of automated tools, often used by "script kiddies"**
- **These attacker tools are capable of probing and scanning a large part of the Internet in a short time period**
  - **Automated attacks use these tools**
  - **Semi-automated (the attacker deploys automated scripts for scanning and compromise of network machines and installation of attack code. Attacker then uses the handler (master) machines to specify the attack type and victim's address**
  - **Manual (the attacker scans machines manually, not used often nowadays)**

Difficulties in Detecting Intrusions

- Attacks Stealthiness
  - Attackers tries to hide their actions from either an individual who is monitoring the system, or an IDS
    - cover their tracks by editing system logs
    - reset a modification date on a file that they replaced modified
- Novel Intrusions
  - Undetectable by signature based IDSs
  - Should be detected as anomalies by observing significant deviations from the normal network behavior
- Distributed/coordinated attack
  - Need for attack correlation

Basic IDS Model



Intrusion Detection Taxonomy

- Information source
  - host-based ID, network-based ID, wireless-network ID, application logs, sensor alerts
- Analysis strategy
  - Anomaly detection vs. misuse detection
    - Data mining approach vs. traditional techniques
- Time aspects in analysis
  - Real-time analysis vs. off-line analysis
- Architecture
  - Single centralized vs. distributed & heterogeneous
- Activeness
  - Active reaction vs. passive reaction
- Continuality
  - Continuous analysis vs. periodic analysis

IDS – Information Source

- **Host-based IDS**
  - **base the decisions on information obtained from a** *single host* **(e.g. system log data, system calls data)**
- **Network-based IDS**
  - **make decisions according to the information and data obtained by** *monitoring the traffic in the network* **to which the hosts are connected**
- **Wireless network IDS**
  - **detect intrusions by analyzing traffic between mobile nodes**
- **Application Logs**
  - **detect intrusions analyzing for example database logs, web logs**
- **IDS Sensor Alerts**
  - **analysis on low-level sensor alarms**
  - **Analysis of alarms generated by other IDSs**

IDS - Analysis Strategy

- ***Misuse detection*** **is based on extensive knowledge of patterns associated with known attacks provided by human experts**
  - **Existing approaches: pattern (signature) matching, expert systems, state transition analysis, data mining**
  - **Major limitations:**
    - **Unable to detect novel & unanticipated attacks**
    - **Signature database has to be revised for each new type of discovered attack**
- ***Anomaly detection*** **is based on profiles that represent normal behavior of users, hosts, or networks, and detecting attacks as significant deviations from this profile**
  - **Major benefit - potentially able to recognize unforeseen attacks.**
  - **Major limitation - possible high false alarm rate, since detected deviations do not necessarily represent actual attacks**
  - **Major approaches: statistical methods, expert systems, clustering, neural networks, support vector machines, outlier detection schemes**

IDS – Time Aspects in Analysis

- Real-time IDS
  - Analyzes the data while the sessions are in progress (e.g. network sessions for network intrusion detection, login sessions for host based intrusion detection)
  - Raises an alarm immediately when the attack is detected
- Off-line IDS
  - Analyzes the data when the information about the sessions are already collected –post-analysis
  - Useful for understanding the attackers' behavior

IDS – Architecture

- **Centralized IDS**
  - **Data analysis is performed in a fixed number of locations, independent of how many hosts are being monitored**
- **Distributed IDS***
  - **Data analysis is performed in a number of locations proportional to the number of hosts that are being monitored**
  - **Necessary for detection of distributed/coordinated attacks targeted at multiple networks/machines**

* Spafford, Zamboni: Intrusion Detection using Autonomous Agents, *Computer Networks*, 2000.

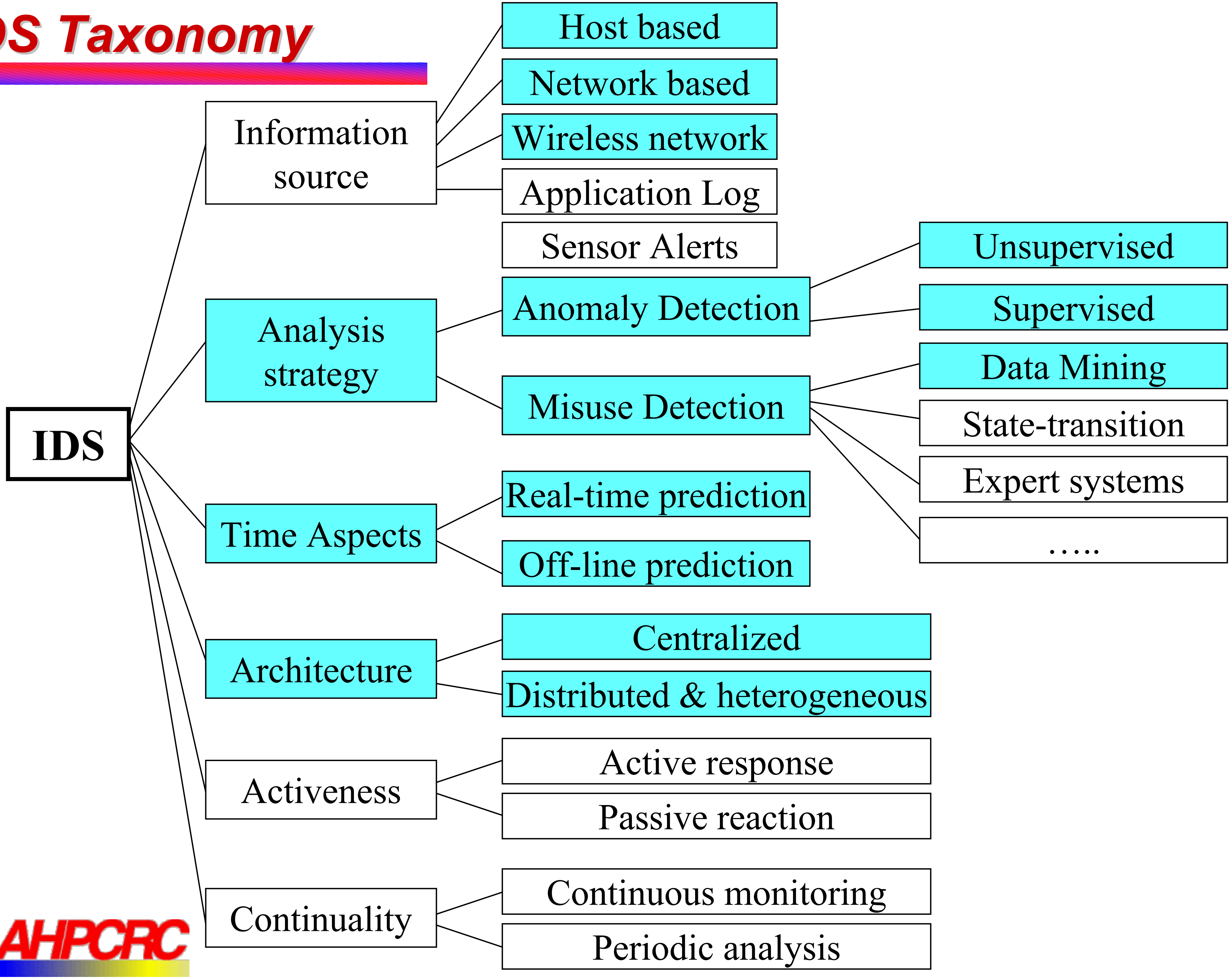
IDS – Activeness

- ***Passive reaction***
  - Merely generates the alarms for the attacks
  - No countermeasure is actively applied to thwart the attack
- ***Active response*** **on attack detection**
  - Corrective response (closing security holes, reconfiguring firewalls, routers and switches…..)
  - Pro-active (logging out attackers, turning off IP addresses terminating network connections
    - NetProbe (cut network connections)
    - CISCO Net Ranger (reconfigures routers and switches, interacts with HP-OpenView
    - Ballista (shutdowns vulnerable services, modifies configuration files, …)
    - …

IDS – Continuality

- Continuous Monitoring
  - IDS performs a continuous, real-time analysis by acquiring information about the actions immediately after they happen
  - Costly process due to transporting the audit data and processing them quickly
- Periodic Analysis
  - IDS periodically takes the snapshot of the environment (monitored system), analyzes the data snapshot looking for vulnerable software or spots and their exploits, configuration errors, etc.
  - Widely used by system administrators, but not satisfactory to ensure high security, since the security exposure between two consecutive runs is sufficient for active exploit of a vulnerability

IDS Taxonomy



Measures for Evaluating IDS

Standard metrics for evaluations of intrusions (attacks)

Standard metrics		Predicted connection label: Normal	Predicted connection label: Intrusions (Attacks)
Actual connection label	Normal	True Negative (TN)	False Alarm (FP)
	Intrusions (Attacks)	False Negative (FN)	Correctly detected intrusions - Detection rate (TP)

- **Standard measures for evaluating IDSs:**
  - ***Detection rate*** **- ratio between the number of correctly detected attacks and the total number of attacks**
  - ***False alarm (false positive)*** **rate - ratio between the number of normal connections that are incorrectly misclassified as attacks (False Alarms in Table) and the total number of normal connections**
  - **Trade-off between detection rate and false alarm rate**
  - **Performance (Processing speed + propagation + reaction)**
  - **Fault Tolerance (resistant to attacks, recovery, resist subversion)**

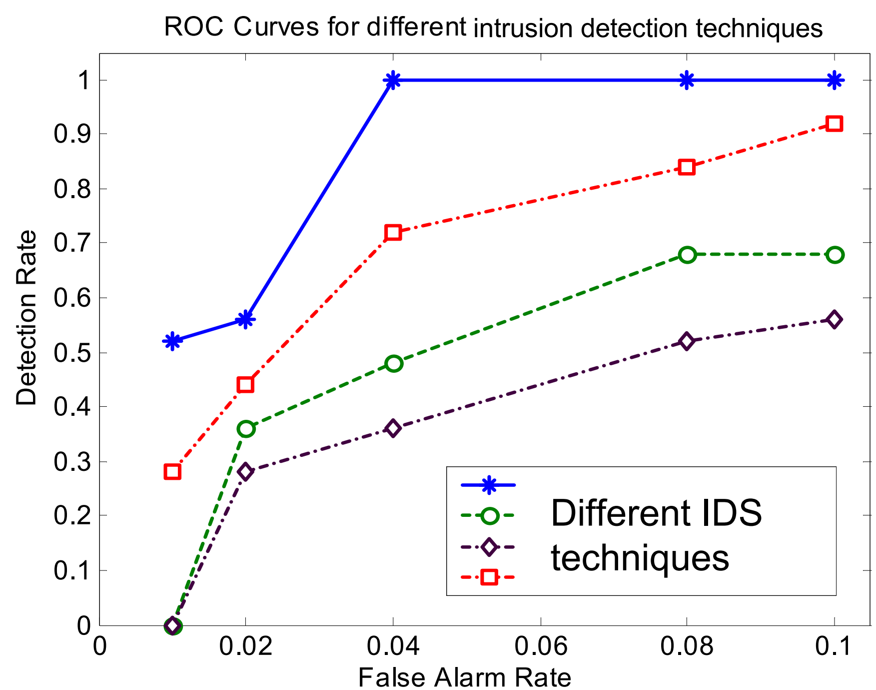
AHPCRC

Measures for Evaluating IDS

Standard metrics for evaluations of intrusions (attacks)

Standard metrics		Predicted connection label	
		Normal	Intrusions (Attacks)
Actual connection label	Normal	True Negative (TN)	False Alarm (FP)
	Intrusions (Attacks)	False Negative (FN)	Correctly detected intrusions - Detection rate (TP)

- **ROC Curves is a trade-off between detection rate and false alarm rate**
- **It is plot for different false alarm rates**
- **Ideal system should have 100% detection rate with 0% false alarm**



Commercial Intrusion Detection Systems

- Current commercial IDSs are largely network-based
- Misuse detection based commercial IDSs
  - SNORT – open source network IDS based on signatures
  - Network Flight Recorder (NFR) detects known attacks and their variations
  - NetRanger (CISCO): sensors (analyze the traffic) and directors (manage sensors)
  - Shadow – collects audit data and runs *tcmdump* filters to catch attacks
  - P-Best (SRI) –rule-based expert system that describes malicious behavior
  - NetStat (UCSB) – real time IDS using state transition analysis
  - …..
- Anomaly detection based commercial IDSs
  - IDES, NIDES – statistical anomaly detection
  - EMERALD - statistical anomaly detection
  - SPADE (Statistical Packet Anomaly Detection Engine) within SNORT
  - Computer Watch (AT&T) – expert system that summarizes security sensitive events and apply rules to detect anomalous behavior
  - Wisdom & Sense – builds a set of rules that statistically describe normal behavior
  - …..

SNORT (www.snort.org)

- SNORT is an open source signature-based Network IDS
- 3 modes SNORT may be configured
  - *sniffer* mode – reads the packets from the network and displays them in a continuous stream on the console
  - *packet logger* mode – logs the packet to the disk
  - *intrusion detection* mode - analyzes network traffic for matches against user defined rules and performs actions based upon what it observes
- SNORT plugin - SPADE automatically detects stealthy port scans
  - SPADE examines TCP-SYN packets and maintains the count of packets observed on (dest IP, dest Port) tuples
  - SPADE checks the probability of every new packet on the (dest IP, dest Port) tuple
  - The lower the probability, the higher the anomaly score
  - Drawback: SPADE raises false alarms on legitimate traffic for which (dest IP, dest Port) combinations are infrequent

	Dest Port			
Dest IP		*		x
		*		
	*	**	***	*
			**	

Data Mining for Intrusion Detection

- ***Misuse detection***
  - Predictive models are built from labeled data sets (instances are labeled as “normal” or “intrusive”)
  - These models can be more sophisticated and precise than manually created signatures
  - Unable to detect attacks whose instances have not yet been observed
- ***Anomaly detection***
  - Build models of “normal” behavior and detect anomalies as deviations from it
  - Possible high false alarm rate - previously unseen (yet legitimate) system behaviors may be recognized as anomalies

Key Technical Challenges

"Mining needle in a haystack.
So much hay and so little time"

- Large data size
  - E.g. Millions of network connections are common for commercial network sites, …
- High dimensionality
  - Hundreds of dimensions are possible
- Temporal nature of the data
  - Data points close in time - highly correlated
- Skewed class distribution
  - Interesting events are very rare $\Rightarrow$ looking for the "needle in a haystack"
- Data Preprocessing
  - Converting data from monitored system into data appropriate for analysis
- High Performance Computing (HPC) is critical for on-line, scalable and distributed intrusion detection

Projects: Data Mining in Intrusion Detection

- **MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection) –** Columbia University, Georgia Tech, Florida Tech
- **ADAM (Audit Data Analysis and Mining) -** George Mason University
- **MINDS (University of Minnesota)**
- **Intelligent Intrusion Detection – IIDS (Mississippi State University)**
- **Data Mining for Network Intrusion Detection (MITRE corporation)**
- **Agent based data mining system (Iowa State University)**
- **IDDM – Department of Defense, Australia**
- **…..**

Data sets in Intrusion Detection

- DARPA 1998[1] data set and its modification KDDCup99 data set created in MADAM ID project
- DARPA 1999[1] data set
- System call traces data set[2] – U. New Mexico
- Solaris audit data using BSM[3] (Basic Security Module)
- University of Melbourne, Australia
  - MOAT – packet trace files
  - Auckland II – packet trace files
- Data set with virus files[4] available from Columbia University

[1] http://www.ll.mit.edu/IST/ideval/data/data_index.html

[2] http://www.cs.unm.edu/~immsec/systemcalls.htm

[3] Sun Microsystems SunShield Basic Security Module Guide.

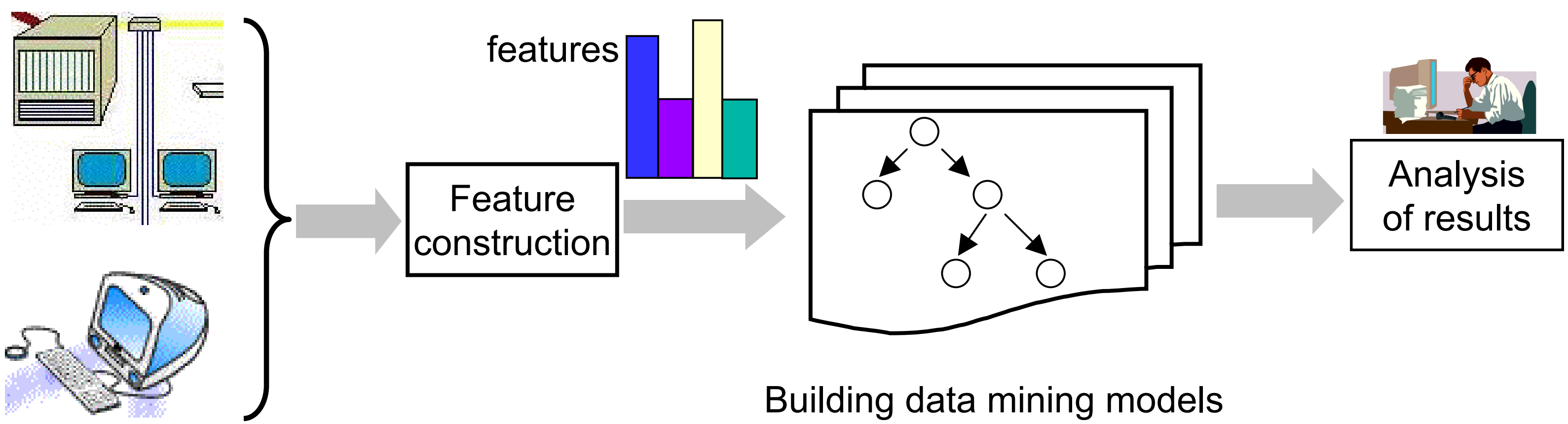
[4] http://www.cs.columbia.edu/ids/mef/software

DARPA 1998 Data Set

- DARPA 1998 data set (prepared and managed by MIT Lincoln Lab) includes a wide variety of intrusions simulated in a military network environment
- 9 weeks of raw TCP dump data
  - 7 weeks for training (5 million connection records)
  - 2 weeks for training (2 million connection records)
- Connections are labeled as normal or attacks (4 main categories of attacks - 38 attack types)
  - DOS - Denial Of Service
  - Probe - e.g. port scanning
  - U2R - unauthorized access to gain root privileges,
  - R2L - unauthorized remote login to machine

Basic steps in Data Mining for ID

- Converting the data from monitored system (computer network, host machine, …) into data (features) that will be used in data mining models
  - For misuse detection, labeling data examples into normal or intrusive may require enormous time for many human experts
- Building data mining models
  - Misuse detection models
  - Anomaly detection models
- Analysis and summarization of results



Feature Construction in Intrusion Detection

- MADAM ID: Convert DARPA'98 data into KDDCup'99 data
- Network traffic data is collected using "sniffers" (e.g. *tcpdump*, *net-flow tools*, …)
- Collected data are in the form of network connections or network packets (a network connection may contain several packets)
- Basic information collected for individual network connections include e.g.
  - start time and duration
  - protocol type
  - source IP address and port,
  - destination IP address and destination port (service)
  - number of bytes, packets in connection
  - …

Feature Construction in Intrusion Detection

MADAM ID Example from network traffic data:

dst …	*service* …	*flag*
h1	http	S0
h1	http	S0
h1	http	S0
h2	http	S0
h4	http	S0
h2	ftp	S0

syn flood

normal

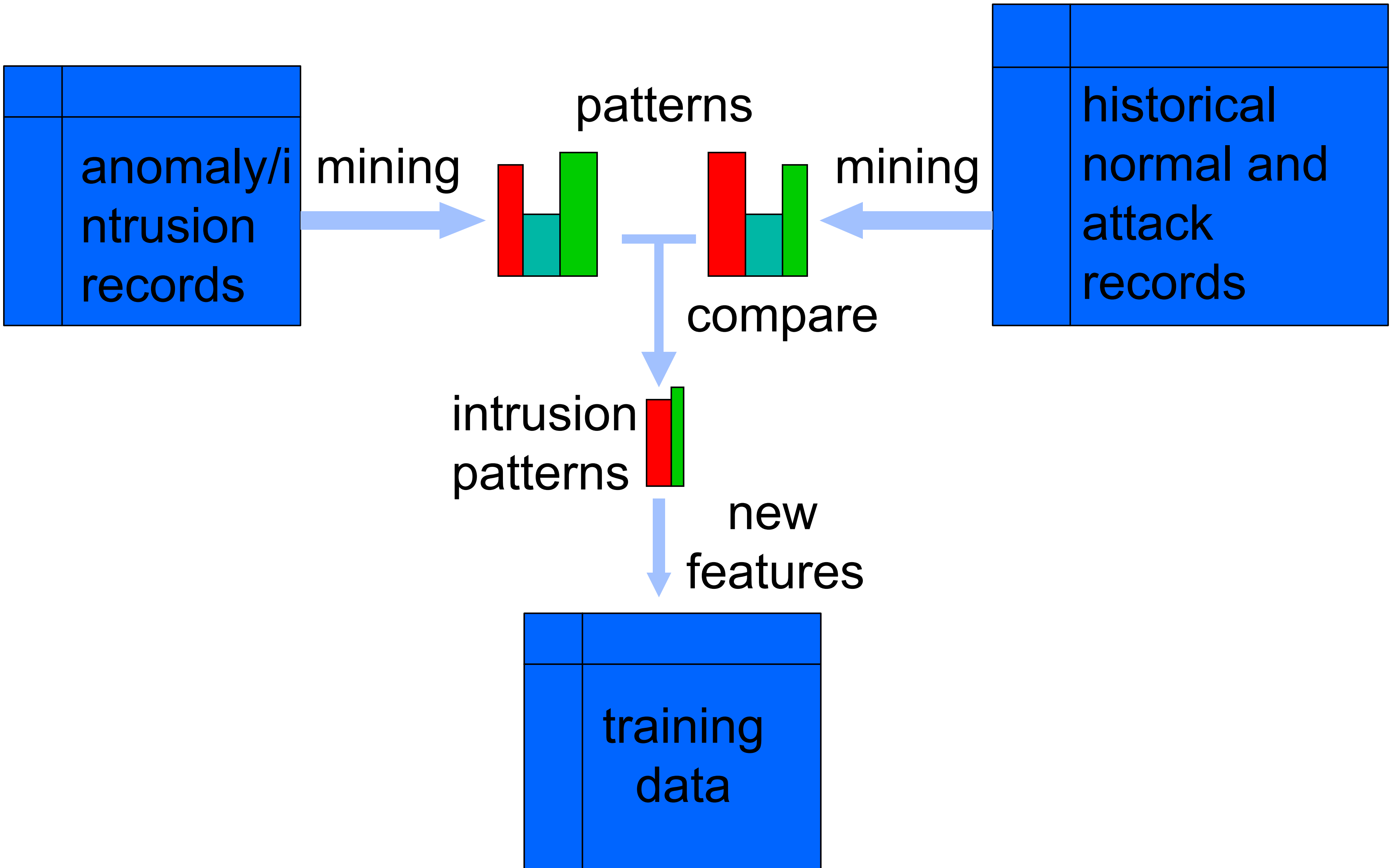
dst …	*service* …	*flag*	%S0
h1	http	S0	70
h1	http	S0	72
h1	http	S0	75
h2	http	S0	0
h4	http	S0	0
h2	ftp	S0	0

Basic existing features may be useless

construct features with high information gain

How? Use temporal and statistical patterns, e.g., "a lot of S0 connections to same service/host within a short time window"

MADAM ID - Feature Construction Example



MADAM ID - Feature Construction Example

- An example: "syn flood" patterns (dst_host is *reference* attribute):
  - (flag = S0, service = http),
    (flag = S0, service = http) $\rightarrow$ (flag = S0, service = http) [0.6, 2s]
  - add features:
    - count the connections to the same *dst_host* in the past 2 seconds, and among these connections,
    - the percentage with the same *service*,
    - the percentage with S0
- Search through the feature space through iterations, at each iteration:
  - Use different heuristics to compute patterns (e.g., per-host service patterns) and construct features accordingly
- Limitations:
  - Connection level only
  - Within-connection contents are not "structured", and much more challenging!

AHPCRC

Feature Construction in Intrusion Detection

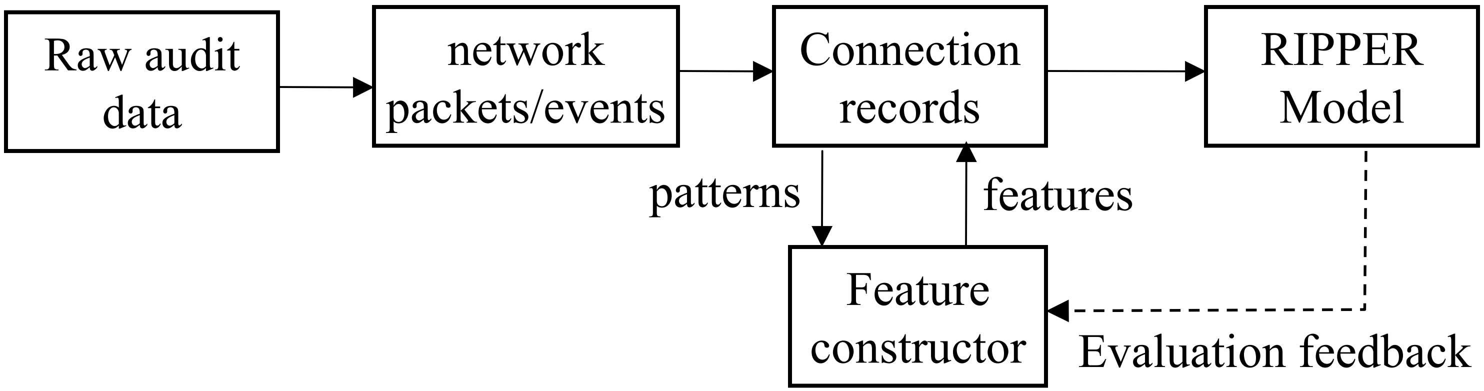
- Three groups of features are constructed (KDDCup 99):
  - "content-based" features within a connection
    - number of packets, acknowledgments, data bytes from *src* to *dest*)
    - Intrinsic characteristics of data packets
  - time-based traffic features included number of connections or different services from the same source or to the same destination considering recent time interval (e.g.a few seconds)
    - Useful for detecting scanning activities
  - connection based features included number of connections from same source or to same destination or with the same service considering in last *N* connections
    - Useful for detecting SLOW scanning activities

AHPCRC

Data Mining for Misuse Detection

- Classification techniques:
  - Rule based techniques (RIPPER, PN-rule, …..)
    - Projects: MADAM ID, ADAM, MINDS, …
  - Tree based approaches (decision trees, similarity trees, …..)
    - MADAM ID, MITRE, …
  - Association rules, fuzzy association rules
    - Projects: MADAM ID, ADAM, MINDS, IIDS
  - Bayesian classifiers, genetic algorithms, LVQ, …..
  - Multiple classifiers (meta-classification, multi Bayes, …..)
  - Neural networks
- Cost sensitive modeling (AdaCost, …..)
- Learning from rare class

MADAM ID Workflow*



- Association rules and frequent episodes are applied to network connection records to obtain additional features for data mining algorithms
- Apply RIPPER to labeled data sets and learn the intrusions

* W. Lee,S. Stolfo, Adaptive Intrusion Detection: a Data Mining Approach, *Artificial Intelligence Review,* 2000

MADAM ID - Cost-sensitive Modeling

- A multiple-model approach:
  - Certain features are more costly to compute than others
  - Build multiple rule-sets, each with features of different cost levels;
  - Use cheaper rule-sets first, costlier ones later only for required accuracy.
- 3 cost levels for features:
  - Level 1: beginning of an event, cost 1;
  - Level 2: middle to end of an event, cost 10;
  - Level 3: multiple events in a time window, cost 100.

AHPCRC

* W. Lee,et al., Toward Cost-Sensitive Modeling for Intrusion Detection and Response, *Journal of Computer Security,* 2002

*ADAM**

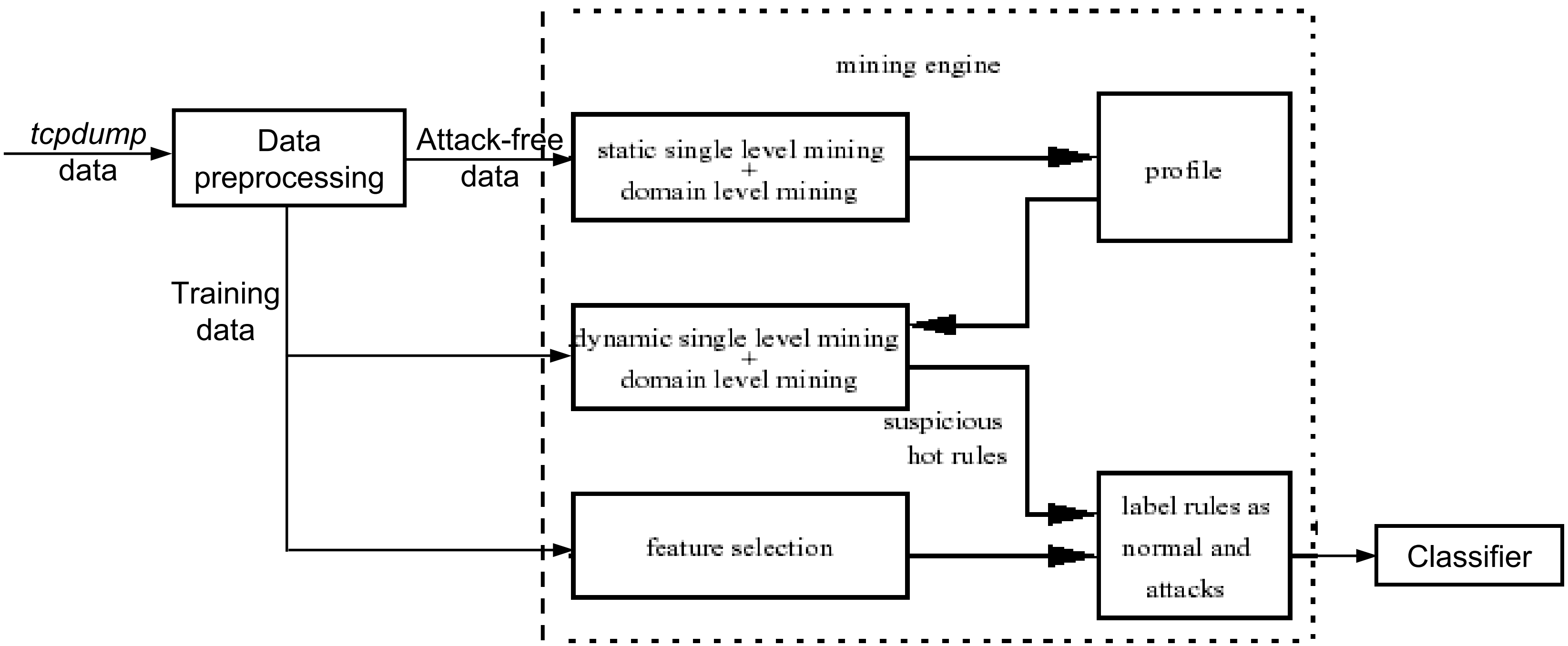
- Data mining testbed that uses combination of association rules and classification to discover attacks
- I phase: ADAM builds a repository of profiles for "normal frequent itemsets" by mining "attack-free" data
- II phase: ADAM runs a sliding window, incremental algorithm that finds frequent itemsets in the last N connections and compare them to "normal" profile

- Tunable: different thresholds can be set for different types of rules.
- Anomaly detection: first characterize normal behavior (profile), then flag abnormal behavior
- Reduced false alarm rate: using a classifier.

* D. Barbara, et al., ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection. SIGMOD Record 2001.

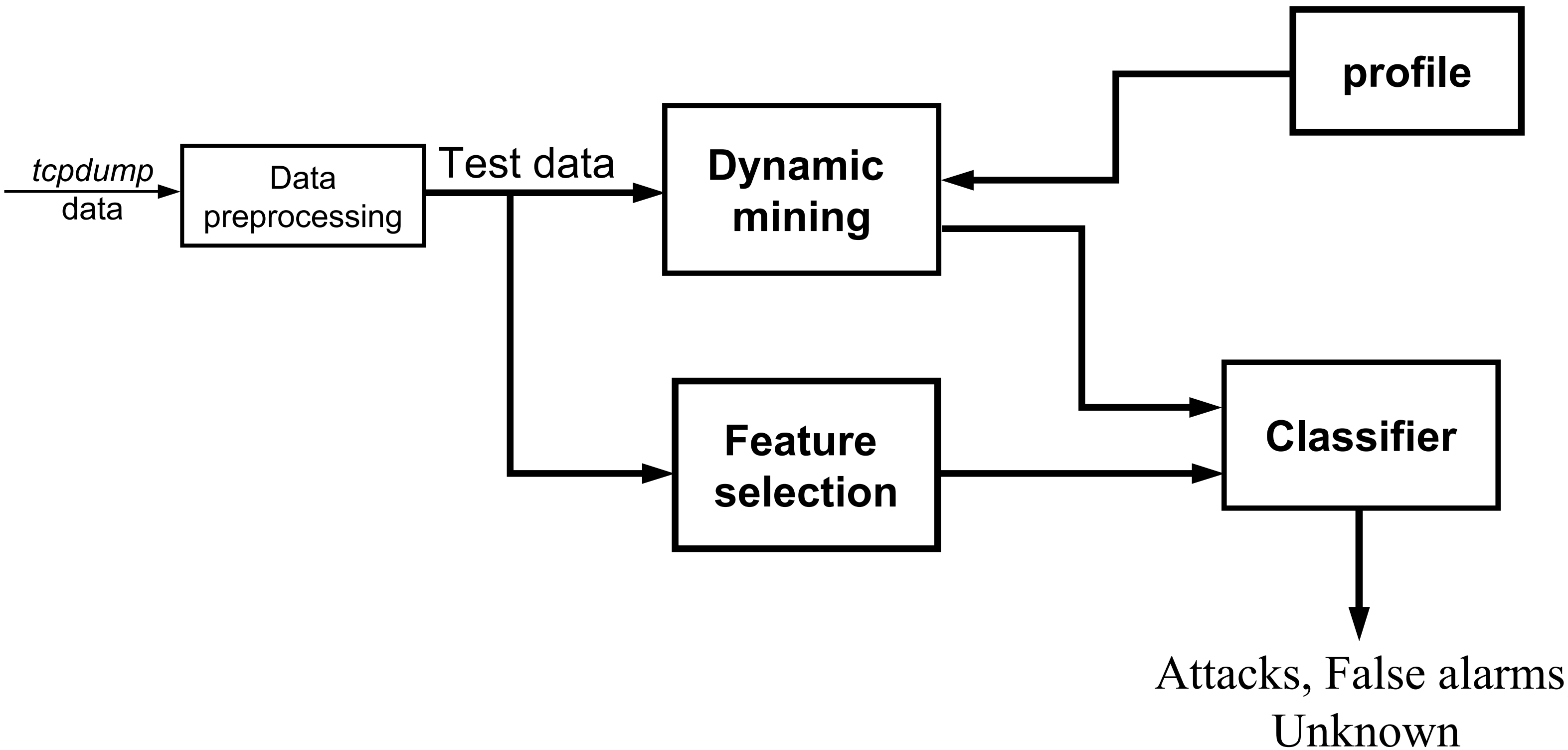
ADAM: Training phase

- Database of frequent itemsets for attack-free data is made
- For entire training data, find suspicious frequent itemsets that are not in the “attack-free” database
- Train a classifier to classify itemset as known attack, unknown attack or normal event



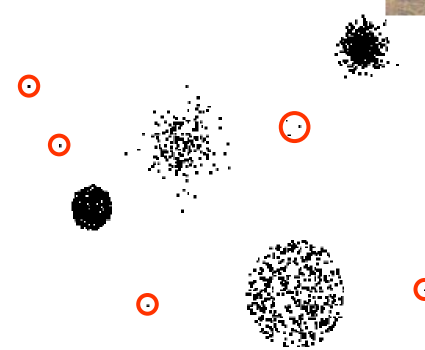
ADAM: Phase of Discovering Intrusions

- **Dynamic mining module produces suspicious itemsets from test data**
- **Along with features from feature selection module, itemsets are fed to classifier**



The MINDS Project

- **MINDS – MINnesota INtrusion Detection System**
  - **Learning from Rare Class – Building rare class prediction models**
  - **Anomaly/outlier detection**
  - **Characterization of attacks using association pattern analysis**



Tid	SrcIP	Start time	Dest IP	Dest Port	Number of bytes
1	206.163.37.95	11:07:20	160.94.179.223	139	192
2	206.163.37.95	11:13:56	160.94.179.219	139	195
3	206.163.37.95	11:14:29	160.94.179.217	139	180
4	206.163.37.95	11:14:30	160.94.179.255	139	199
5	206.163.37.95	11:14:32	160.94.179.254	139	186
6	206.163.37.95	11:14:35	160.94.179.253	139	177
7	206.163.37.95	11:14:36	160.94.179.252	139	172
8	206.163.37.95	11:14:38	160.94.179.251	139	192
9	206.163.37.95	11:14:41	160.94.179.250	139	195
10	206.163.37.95	11:14:44	160.94.179.249	139	163

Rules Discovered:

{Src IP = 206.163.37.95,
Dest Port = 139,
Bytes ∈ [150, 200]} --> {SCAN}

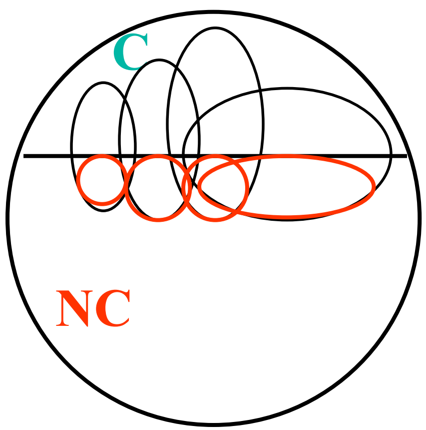
AHPCRC

MINDS - Learning from Rare Class

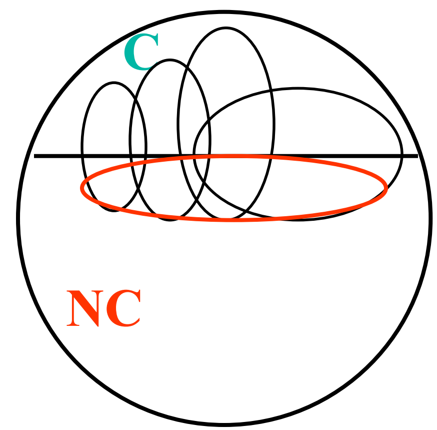
- Problem: Building models for rare network attacks (Mining needle in a haystack)
  - Standard data mining models are not suitable for rare classes
    - Models must be able to handle skewed class distributions
  - Learning from data streams - intrusions are sequences of events
- Key results:
  - *PNrule* and related work
  - *Boosting based algorithms (RareBoost , SMOTEBoost)*
  - CREDOS algorithm
  - Classification based on association - add frequent items as “meta-features” to original data set

PN-rule Learning and Related Algorithms*

- *P-phase*:
  - **cover most of the positive examples with high support**
  - **seek good recall**
- *N-phase*:
  - **remove FP from examples covered in P-phase**
  - **N-rules give high accuracy and significant support**



Existing techniques can possibly learn erroneous small signatures for absence of C



PNrule can learn strong signatures for presence of NC in *N-phase*

* M. Joshi, et al., PNrule, Mining Needles in a Haystack: Classifying Rare Classes via Two-Phase Rule Induction, ACM SIGMOD 2001

Boosting based algorithms

- **RareBoost ***
  - **updates the weights of the examples differently for false positives, false negatives, true positives and true negatives**
- **SMOTEBoost ****
  - **SMOTE (*Synthetic Minority Oversampling Technique*) generates artificial examples from minority (rare) class along the boundary line segment**
    - **Generalization of over-sampling technique**
  - **Combination of SMOTE and boosting further improves the prediction performance or rare classes**

* M. Joshi, et al, Predicting Rare Classes: Can Boosting Make Any Weak Learner Strong?, ACM SIGKDD 2002.

** A. Lazarevic, et al, SMOTEBoost: Improving the Prediction of Minority Class in Boosting, in review.

CREDOS*

- CREDOS is a novel algorithm that first uses the ripple down rules to overfit the training data and then to prune them to improve generalization capability
- Ripple down rules (RDRs) are commonly used since they make knowledge bases easy and efficient to use and maintain, but these induction algorithms do not have a pruning phase (prone to overfitting)
- Different pruning mechanism than the ones used for decision trees is required due to their unique structure
- In CREDOS, a generic pruning framework and its specific MDL-based version is used.

* M. Joshi, V. Kumar, CREDOS: Classification using Ripple Down Structure (A Case for Rare Classes), ICDE 2003

Alternative Classification Approaches

- Fuzzy Data Mining in network intrusion detection (MSU)*
- Create fuzzy association rules only from normal data to learn "normal behavior"
  - For new audit data, create the set of fuzzy association rules and compute its similarity to the "normal" one
  - If similarity low $\Rightarrow$ for new data generate alarm
- Genetic algorithms (GA) used to tune membership function of the fuzzy sets
  - Fitness – rewards for high similarity between normal and reference data, penalizing – high similarity between intrusion and reference data
- Use GA to select most relevant features

AHPCRC

* S. Bridges, R. Vaughn, Intrusion Detection Via Fuzzy Data Mining, 2000

Alternative Classification Approaches

- **Detection of New Viruses***
- **Current virus scan techniques are signature based (e.g. VirusScan, …)**
- **Apply standard data mining algorithms on a set of malicious (virus) and benign executables using the derived features:**
  - **List of used DLLs, DLL function calls and their number**
  - **Headers of programs contain strings and each string was used as a feature**
  - ***hexdump* binary files into hexadecimal files, and each byte sequence was used as a feature**
- **Apply RIPPER, Naive Bayes, multi-Bayes classifiers to detect malicious virus code**

* M. Schultz, et al., Data Mining Methods for Detection of New Malicious Executables, IEEE Symposium on Security 2001.

Alternative Classification Approaches*

- Decision trees*
- Simple application of ID3 using basic host session records (network data)
  - Use Genetic Algorithms to create rules that match anomalous connection
    - Fitness – actual performance on pre-classified data
    - Use nitching techniques to create multiple rules for different types of anomalies

* C. Sinclair, L. Pierce, S. Matzner: An Application of Machine Learning to Network Intrusion Detection , 1998.

Alternative Classification Approaches*

- *Hybrid Approach to Profile Creation**
- **Assumption: Over time users establish profile based on the number and types of commands**
  - **attributes are percentage of commands used by user**
- **Methodology:**
  - **Reduce dimensionality using expert rules**
  - **Cluster data using *k-means* clustering**
  - **Further reduce dimensionality using Genetic Algorithms**
  - **Refine the cluster locations using LVQ (Linear Vector Quantization)**
    - **Nearest neighbor classifier based on SOMs (Self Organizing Maps)**

* J. Marin, D. Ragsdale, J. Surdu: A Hybrid Approach to the Profile Creation and Intrusion Detection , 2001.

Alternative Classification Approaches*

- Scalable Clustering Technique*
- Apply supervised clustering
  - For each point find nearest point and if belong to the same class, append to the same cluster, else create a new
- Classification
  - Class dominated in k nearest clusters
  - Weighted sum of distances to k nearest clusters
- Incremental clustering

- Distances: weighted Euclidean, Chi-square, Canbera

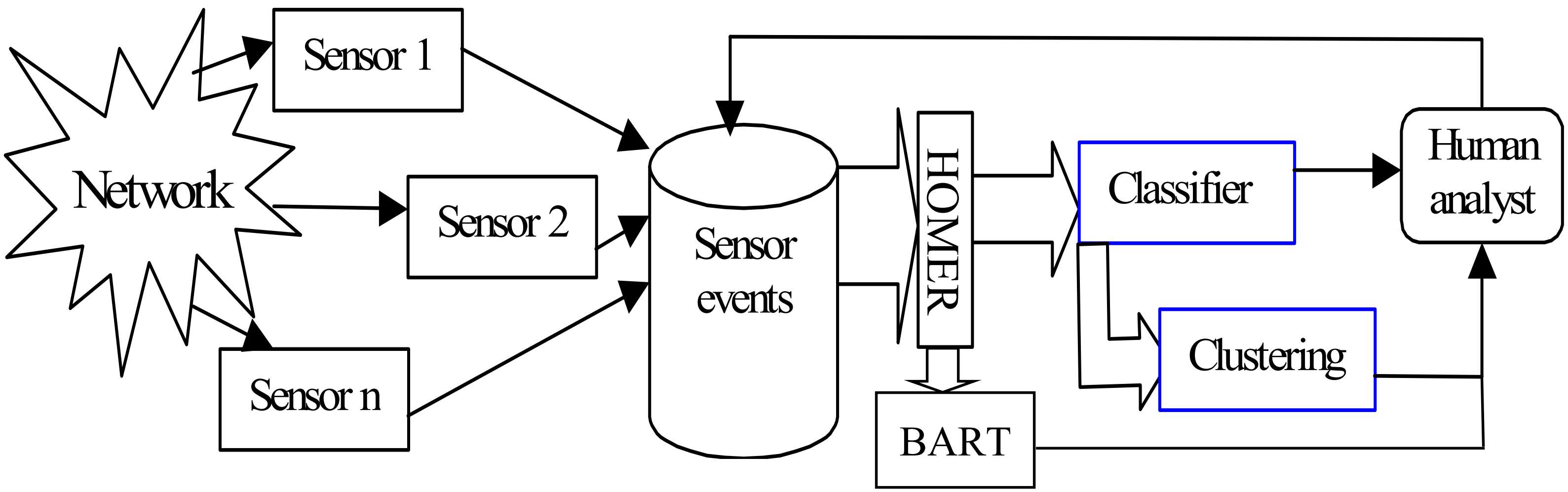
$$(d(x, L) = \sum_{i=1}^{d} \frac{|x_i - L_i|}{x_i + L_i} C_i^2$$

AHPCRC

* N. Ye, X. Li, A Scalable Clustering for Intrusion Signature Recognition, 2001.

Alternative Classification Approaches*

- **MITRE* methods are used to:**
  - **Promote interesting results**
  - **Aggregate ‘similar’ events**
  - **Demote uninteresting results**



AHPCRC

* E. Bloedorn, et al., Data Mining for Network Intrusion Detection: How to Get Started, 2001.

MITRE: Promoting interesting alerts

- Goal: Identify those network events most likely to be the greatest security threat
- Methods:
  - Use classification tree or rule methods to describe categories of events (e.g. ‘interesting/uninteresting’)
  - Use regression to find importance of dependent attributes on ‘interestingness’ metric
  - Use clustering methods to identify small or distant clusters of events
  - Use outlier detection methods like attribute focusing, Gritbot
  - Use statistics to identify outliers along a single dimension (BART)

MITRE: Uninteresting Events and Aggregating Alerts

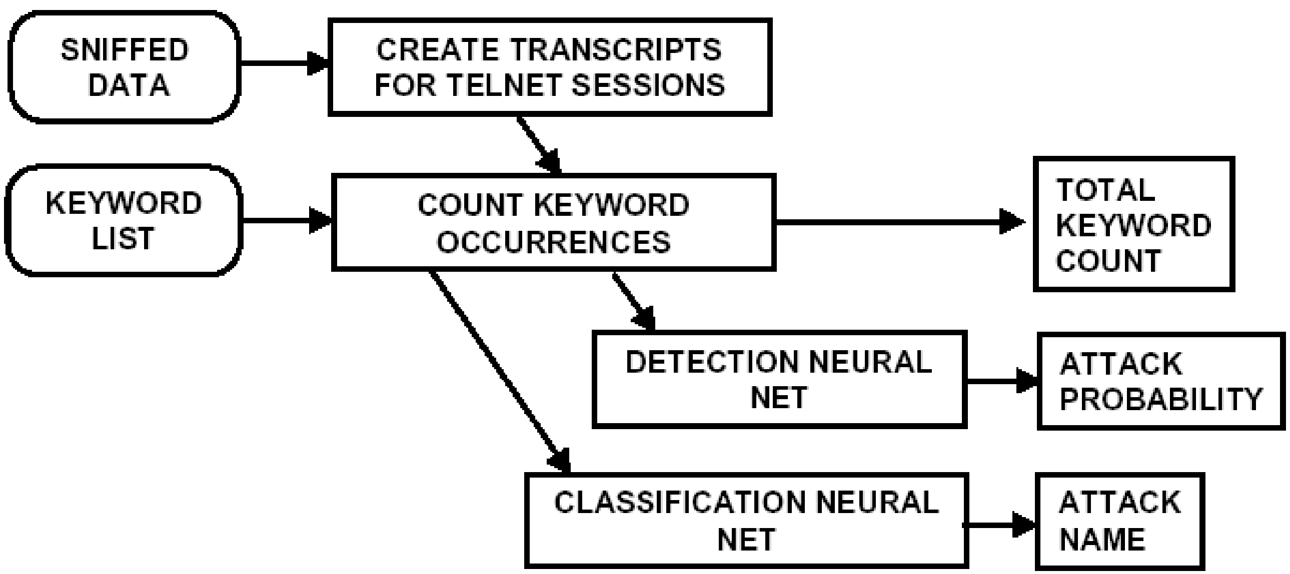
- **Demote Uninteresting Events: Identify those network events least likely to be the greatest security threat**
- **Methods:**
  - **Use classification tree or rule methods to describe categories of events (e.g. ‘interesting/<u>uninteresting</u>’)**
  - **Use sequential association rules to characterize most common sequences**
  - **Use clustering methods to identify large clusters of events**
- **Aggregating related alerts together is used to reduce the number of alerts shown to analyst Methods:**
  - **Use sequential associations to find near identical events in a short time window**
  - **Use domain knowledge and statistics to build models of IP-mapping and Port-scanning (HOMER, GHOST)**

Neural Networks Classification Approaches

- Neural networks (NNs) are applied to host-based intrusion detection
  - Building profiles of users according to used commands
  - Building profiles of software behavior
- Neural networks for network-based intrusion detection
  - Hierarchical network intrusion detection
  - Multi-layer perceptrons (MLP)
  - Self organizing maps (SOMs)

Using Keyword Selection and NNs*

- Data set: Unix environment
- Generic keywords are selected to detect attack preparation, the actual break-in and action after break-in
- Keywords that users use in Unix are used as attributes
- NNs are used to learn between normal and anomalies.

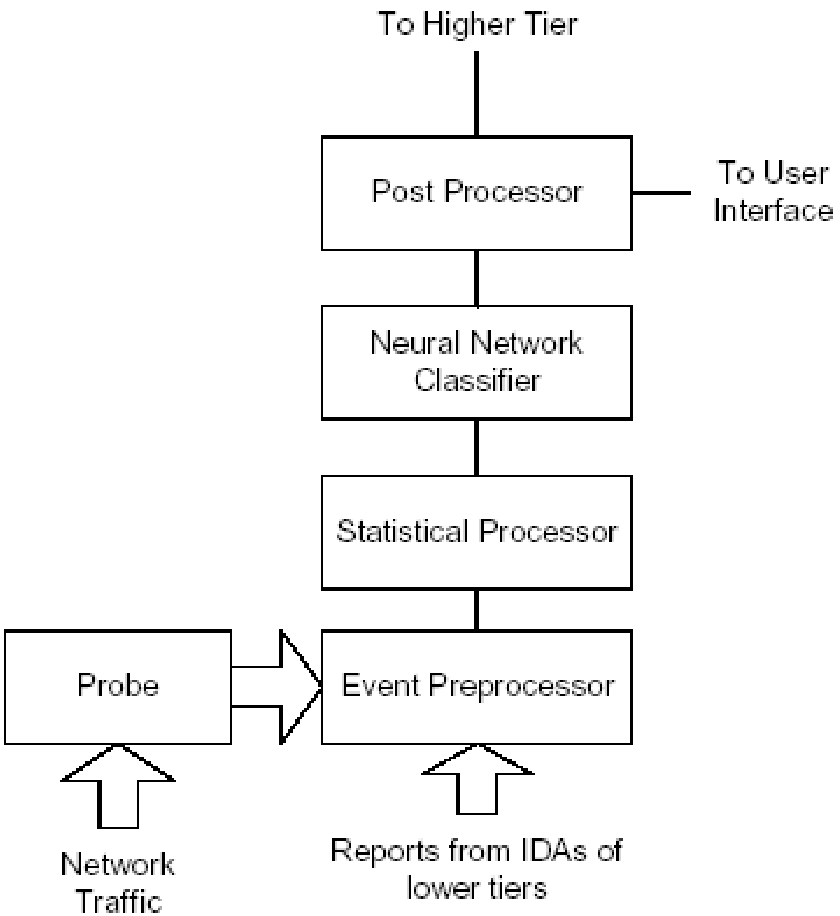


AHPCRC

* R. Lippmann, R. Cunningham, Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks, 2000

Hierarchical NID using Neural Networks*

- Applied to network intrusion detection
- Architecture
  - Probe: collects the network traffic
  - Event preprocessor: receives the reports from probe and IDAs and converts the information into the format for statistical model
  - Statistical processor: Maintains a reference model of typical network activities, compares it to the data from event preprocessor and forms a vector to feed into NN
  - Neural Network classification
  - Post processor – generate reports



*Z. Zheng, et al, HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification, 2001.

NNs for Misuse Detection*

- Data collected by RealSecure™ network monitor
  - 9 basic attributes (protocol, souce & destination ID, …)
- Use multi-layer perceptron (MLP) to learn attacks
- For temporally dispersed and distributed attacks
  - Use SOM to categorize events and forward corresponding number to
  - MLP that classifies normal/attack data

AHPCRC

* J. Canady, J. Mahaffey, The Application of Artificial Neural Networks to Misuse Detection:Initial Results, 1998.

NNs for Profiling Authorized Users*

- Detecting intruders logging into a computer network in the Unix OS environment
- Each user is characterized by
  - input data: unique characteristics of user logging into a network – command, host, time and execution time
  - Output data: authorized users or intruders
- Apply different NN models to detect intruders

AHPCRC

* V. Dao, R. Vemuri, A Performance Comparison of Different Back Propagation Neural Networks Methods in Computer Network Intrusion Detection, 2000

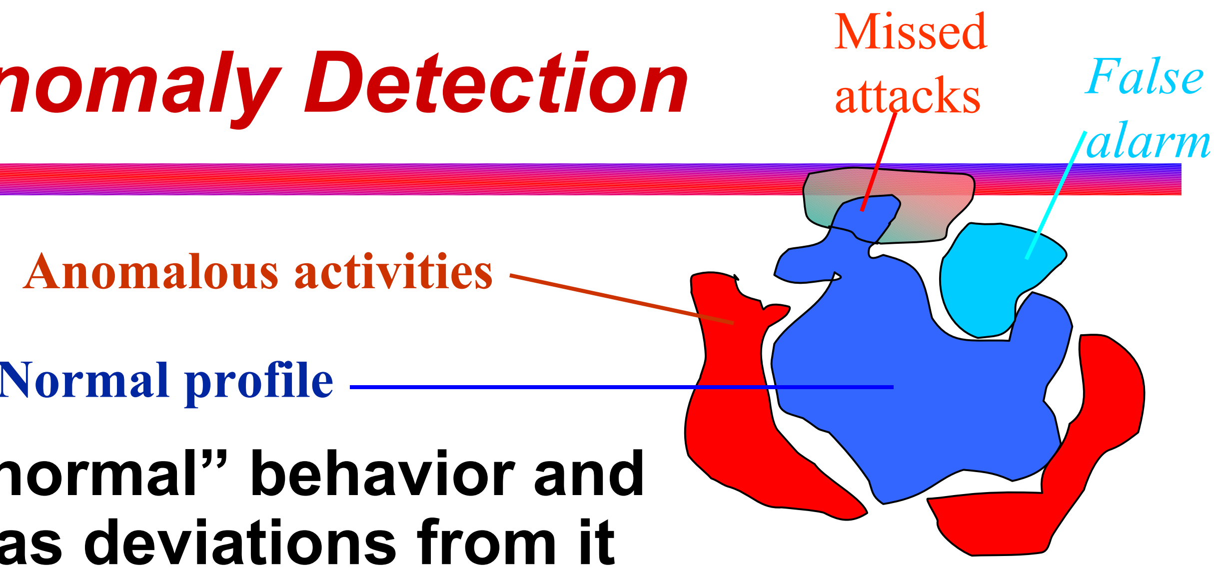
Classification: Cost Sensitive Modeling*

- Statistical accuracy , Detection rate/False Alarm rate may be misleading $\Rightarrow$ *cost based metrics*
- Cost factors: damage cost, response cost, operational cost (level 1-4 features - use cheaper rule-sets first, costlier ones later only for required accuracy
- Costs for TP, FP, TN, FN
- Define cumulative cost
- Apply AdaCost: misclassified examples are weighted by statistical accuracy and by the "cost"
  - elements that have higher cost - higher chance of being selected

AHPCRC

* W. Fan, S. Stolfo, J. Zhang, and P. Chan, Adacost: Misclassification cost-sensitive boosting, ICML 1999.

Data Mining - Anomaly Detection

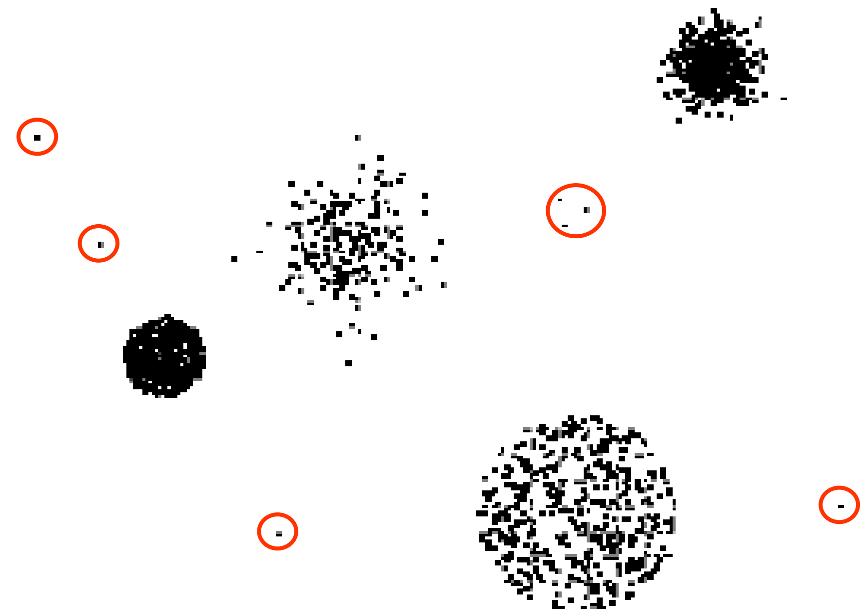


- Build models of "normal" behavior and detect anomalies as deviations from it
- Possible high false alarm rate - previously unseen (yet legitimate) system behaviors may be recognized as anomalies
- Major approaches:
  - Outlier detection
  - Profiling based techniques
  - Other techniques
- Two types of techniques
  - with access to normal data
  - with NO access to normal data (not known what is "normal")

AHPCRC

Outlier Detection Schemes

- Outlier is defined as a data point which is very different from the rest of the data based on some measure
- Detect novel attacks/intrusions by identifying them as deviations from "normal", i.e. anomalous behavior
  - Identify normal behavior
  - Construct useful set of features
  - Define similarity function
  - Use outlier detection algorithm
    - Statistics based approaches
    - Distance based approaches
      - Nearest neighbor approaches
      - Clustering based approaches
      - Density based schemes
    - Model based schemes



AHPCRC

Statistics Based Outlier Detection Schemes

- Statistics based approaches – data points are modeled using stochastic distribution $\Rightarrow$ points are determined to be outliers depending on their relationship with this model
  - With high dimensions, difficult to estimate distributions
- Major approaches
  - Finite Mixtures
  - BACON
  - Using probability distribution
  - Information Theory measures

Statistics Based Outlier Detection Schemes

- Using Finite Mixtures – SmartSifter (SS)*
- SS uses a probabilistic model as a representation of underlying mechanism of data generation.
  - Histogram density used to represent a probability density for categorical attributes
    - SDLE for learning histogram density for categorical domain
  - Finite mixture model used to represent a probability density for continuous attributes
    - SDEM for learning finite mixture for continuous domain
- SS gives a score to each example x_i on the basis of the learned model, measuring how large the model has changed after the learning

AHPCRC

* K. Yamanishi, On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, KDD 2000

Statistics Based Outlier Detection Schemes

- **Using Probability Distributions***
- **Basic Assumption: # of normal elements in the data is significantly larger then # of anomalies**
- **Distribution for the data *D* is given by:**
  - $D = (1-\lambda)\cdot M + \lambda\cdot A$
    M **- majority distribution,** A **- anomalous distribution**
  - M_t, A_t **sets of normal, anomalous elements respectively**
  - **Compute likelihood $L_t(D)$ of distribution D at time t**
  - **Measure how likely each element x_t is outlier:**
    - $M_t = M_{t-1} \setminus \{x_t\}$, $A_t = A_{t-1} \cup \{x_t\}$
    - **Measure the difference ($L_t - L_{t-1}$)**

* E. Eskin, Anomaly Detection over Noisy Data using Learned Probability Distributions, ICML 2000

Statistics Based Outlier Detection Schemes

- ***Using Information-Theoretic Measures****
- **Entropy measures the uncertainty (impurity) of data items**
  - The entropy is smaller when the class distribution is skewer
  - Each *unique* data record represents a class => the smaller the entropy the fewer the number of different records (higher redundancies)
  - If the entropy is large, data is partitioned into *more regular* subsets
  - Any deviation from achieved entropy indicates potential intrusion
  - Anomaly detector constructed on data with smaller entropy will be simpler and more accurate
- **Conditional entropy H(X|Y) tells how much uncertainty remains in sequence of events X after we have seen subsequence Y ($Y \in X$)**
- **Relative Conditional Entropy**

AHPCRC

* W. Lee, et al, Information-Theoretic Measures for Anomaly Detection, IEEE Symposium on Security 2001

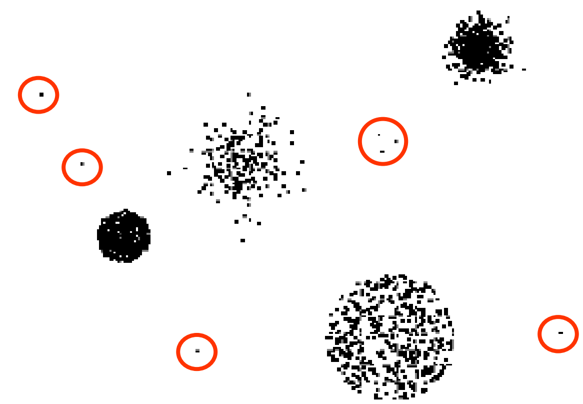
Statistics Based Outlier Detection Schemes

- ***Packet level (PHAD) and Application level (ALAD) anomaly detection****
- **PHAD (packet header anomaly detection) monitors Ethernet, IP and transport layer packet headers**
  - **It builds profiles for 33 different fields from these headers by looking attack free traffic and clustering (prespecifed # of clusters)**
  - **A new value that does not fit into any of the clusters, it is treated as a new cluster and closest two clusters are merged**
  - **The number of updates, *r*, is maintained for each field as well as the number of observations, *n***
  - **Testing: For each new observed packet, if the value for some attribute does not fit into the clusters, anomaly score for that attribute is proportional to *n/r***
- **ALAD uses the same method for anomaly scores, but it works only on TCP data and build TCP streams**
  - **It build profiles for 5 different features**

AHPCRC

* M. Mahoney, P. Chan: Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks, 8th ACM KDD, 2002

Distance based outlier detection schemes

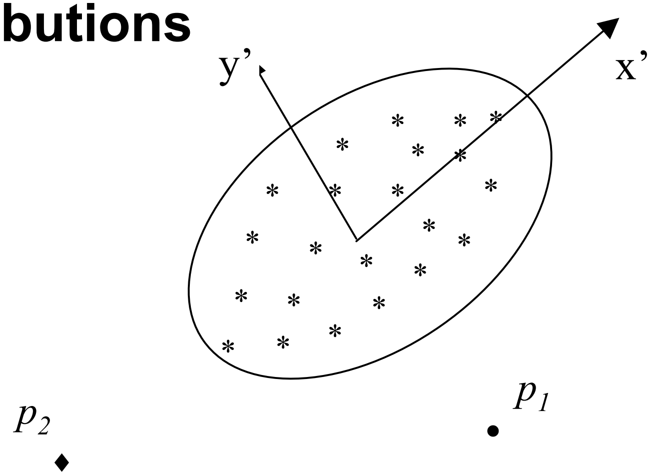


- **Nearest neighbor based approaches (*NN approach*) - Outliers are points that do not have enough neighbors**
- **Density based approach (*LOF approach*) finds outliers based on the densities of local neighborhoods**
  - **Concept of locality becomes difficult to define due to data sparsity in high dimensional space**
- **Clustering based approaches define outliers as points which do not lie in clusters**
  - **Implicitly define outliers as background noise or very small clusters**

AHPCRC

Distance based Outlier Detection Schemes

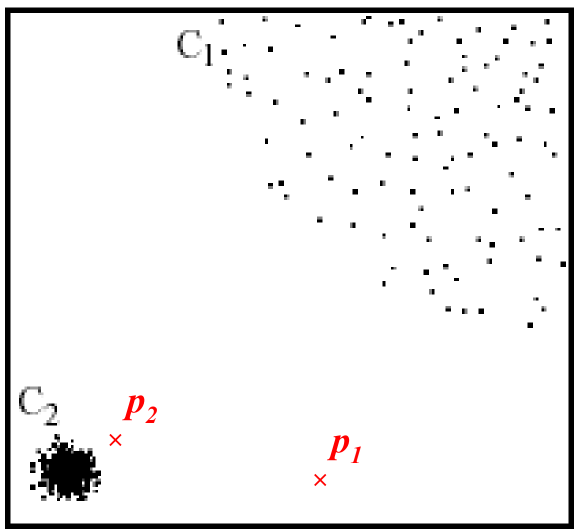
- ***Nearest Neighbor (NN) approach*** [1, 2]
  - **For each point compute the distance to the *k-th* nearest neighbor d_k**
  - **Outliers are points that have larger distance d_k and therefore are located in the more sparse neighborhoods**
- ***Mahalanobis-distance based approach***
  - **Mahalanobis distance is more appropriate for computing distances with skewed distributions**



1. Knorr, Ng,Algorithms for Mining Distance-Based Outliers in Large Datasets, VLDB98
2. S. Ramaswamy, R. Rastogi, S. Kyuseok: Efficient Algorithms for Mining Outliers from Large Data Sets, ACM SIGMOD Conf. On Management of Data, 2000.

Density based Outlier Detection Schemes

- *Local Outlier Factor (LOF) approach* *
  - For each point compute the density of local neighborhood
  - Compute *LOF* of example *p* as the average of the ratios of the density of example *p* and the density of its nearest neighbors
  - Outliers are points with the largest *LOF* value



In the *NN* approach, p_2 is not considered as outlier, while the *LOF* approach find both p_1 and p_2 as outliers

*- Breunig, et al, LOF: Identifying Density-Based Local Outliers, KDD 2000.

* - A. Lazarevic, et al., A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection, SIAM 2003

*Summarization of Anomalous Connections**

- **MINDS example: January 26, 2003 (48 hours after the Slammer worm)**

score	srcIP	sPort	dstIP	dPort	protocol	flags	packets	bytes
37674.69	63.150.X.253	1161	128.101.X.29	1434	17	16	[0,2)	[0,1829)
26676.62	63.150.X.253	1161	160.94.X.134	1434	17	16	[0,2)	[0,1829)
24323.55	63.150.X.253	1161	128.101.X.185	1434	17	16	[0,2)	[0,1829)
21169.49	63.150.X.253	1161	160.94.X.71	1434	17	16	[0,2)	[0,1829)
19525.31	63.150.X.253	1161	160.94.X.19	1434	17	16	[0,2)	[0,1829)
19235.39	63.150.X.253	1161	160.94.X.80	1434	17	16	[0,2)	[0,1829)
17679.1	63.150.X.253	1161	160.94.X.220	1434	17	16	[0,2)	[0,1829)
8183.58	63.150.X.253	1161	128.101.X.108	1434	17	16	[0,2)	[0,1829)
7142.98	63.150.X.253	1161	128.101.X.223	1434	17	16	[0,2)	[0,1829)
5139.01	63.150.X.253	1161	128.101.X.142	1434	17	16	[0,2)	[0,1829)
4048.49	142.150.Y.101	0	128.101.X.127	2048	1	16	[2,4)	[0,1829)
4008.35	200.250.Z.20	27016	128.101.X.116	4629	17	16	[2,4)	[0,1829)
3657.23	202.175.Z.237	27016	128.101.X.116	4148	17	16	[2,4)	[0,1829)
3450.9	63.150.X.253	1161	128.101.X.62	1434	17	16	[0,2)	[0,1829)
3327.98	63.150.X.253	1161	160.94.X.223	1434	17	16	[0,2)	[0,1829)
2796.13	63.150.X.253	1161	128.101.X.241	1434	17	16	[0,2)	[0,1829)
2693.88	142.150.Y.101	0	128.101.X.168	2048	1	16	[2,4)	[0,1829)
2683.05	63.150.X.253	1161	160.94.X.43	1434	17	16	[0,2)	[0,1829)
2444.16	142.150.Y.236	0	128.101.X.240	2048	1	16	[2,4)	[0,1829)
2385.42	142.150.Y.101	0	128.101.X.45	2048	1	16	[0,2)	[0,1829)
2114.41	63.150.X.253	1161	160.94.X.183	1434	17	16	[0,2)	[0,1829)
2057.15	142.150.Y.101	0	128.101.X.161	2048	1	16	[0,2)	[0,1829)
1919.54	142.150.Y.101	0	128.101.X.99	2048	1	16	[2,4)	[0,1829)
1634.38	142.150.Y.101	0	128.101.X.219	2048	1	16	[2,4)	[0,1829)
1596.26	63.150.X.253	1161	128.101.X.160	1434	17	16	[0,2)	[0,1829)
1513.96	142.150.Y.107	0	128.101.X.2	2048	1	16	[0,2)	[0,1829)
1389.09	63.150.X.253	1161	128.101.X.30	1434	17	16	[0,2)	[0,1829)
1315.88	63.150.X.253	1161	128.101.X.40	1434	17	16	[0,2)	[0,1829)
1279.75	142.150.Y.103	0	128.101.X.202	2048	1	16	[0,2)	[0,1829)
1237.97	63.150.X.253	1161	160.94.X.32	1434	17	16	[0,2)	[0,1829)
1180.82	63.150.X.253	1161	128.101.X.61	1434	17	16	[0,2)	[0,1829)
1107.78	63.150.X.253	1161	160.94.X.154	1434	17	16	[0,2)	[0,1829)

Potential Rules:

1.

{Dest Port = 1434/UDP #packets ∈ [0, 2)} --> Highly anomalous behavior (Slammer Worm)

2.

{Src IP = 142.150.Y.101, Dest Port = 2048/ICMP #bytes ∈ [0, 1829]} --> Highly anomalous behavior (ping – scan)

* - Ertoz, L., Eilertson, E., Lazarevic, et al, The MINDS - Minnesota Intrusion Detection System, review for the book "Data Mining: Next Generation Challenges and Future Directions", AAAI/MIT Press.

*Clustering based outlier detection schemes**

- **Radius ω of proximity is specified**
- **Two points x_1 and x_2 are "near" if $d(x_1, x_2) \leq \omega$**
- **Define $N(x)$ – number of points that are within ω of x**
- **Time Complexity $O(n^2)$ $\Rightarrow$ approximation of the algorithm**
- **Fixed-width clustering is first applied**
  - **The first point is a center of a cluster**
  - **If every subsequent point is "near" add to a cluster**
    - **Otherwise create a new cluster**
  - **Approximate N(x) with N(c)**
  - **Time Complexity – $O(cn)$, c - # of clusters**
- **Points in small clusters - anomalies**

* E. Eskin et al., A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, 2002

Clustering based outlier detection schemes

- K-nearest neighbor + canopy clustering approach *
- **Compute the sum of distances to the k nearest neighbors (*k*-NN) of each point**
  - **Points in dense regions – small k-NN score**
  - ***k* has to exceed the frequency of any given attack type**
  - **Time complexity $O(n^2)$**
- **Speed up with *canopy clustering* that is used to split the entire space into small subsets (canopies) and then to check only the nearest points within the canopies**
- **Apply fixed width clustering and compute distances within clusters and to the centers of other clusters**

* E. Eskin et al., A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, 2002

Clustering based outlier detection schemes

- FindOut algorithm* by-product of *WaveCluster*
- **Main idea: Remove the clusters from original data and then identify the outliers**
- **Transform data into multidimensional signals using wavelet transformation**
  - **High frequency of the signals correspond to regions where is the rapid change of distribution – boundaries of the clusters**
  - **Low frequency parts correspond to the regions where the data is concentrated**
- **Remove these high and low frequency parts and all remaining points will be outliers**

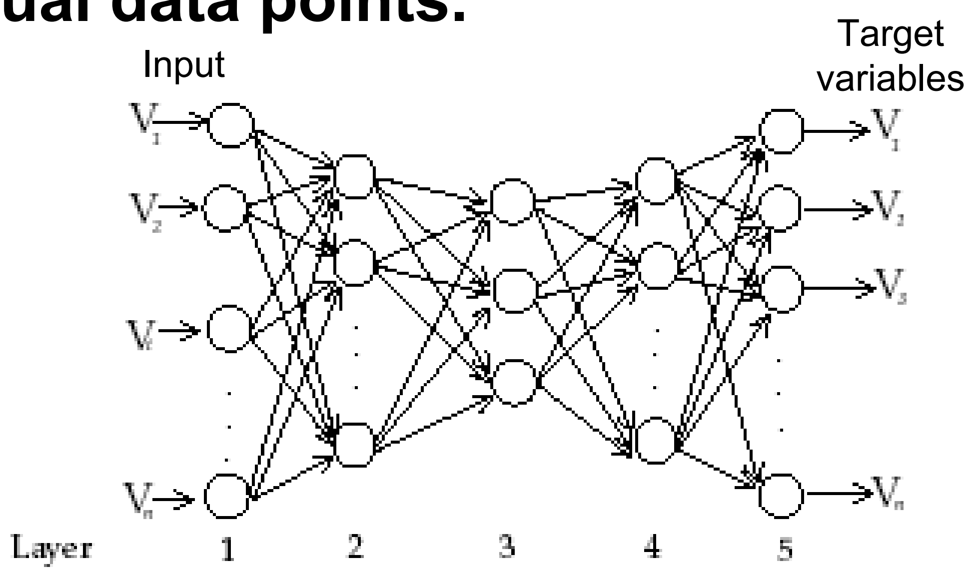
* E. Eskin et al., A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, 2002

Model based outlier detection schemes

- **Use a prediction model to learn the normal behavior**
- **Every deviation from learned prediction model can be treated as anomaly or potential intrusion**
- **Recent approaches:**
  - **Neural networks**
  - **Unsupervised Support Vector Machines (SVMs)**

*Neural networks for outlier detection**

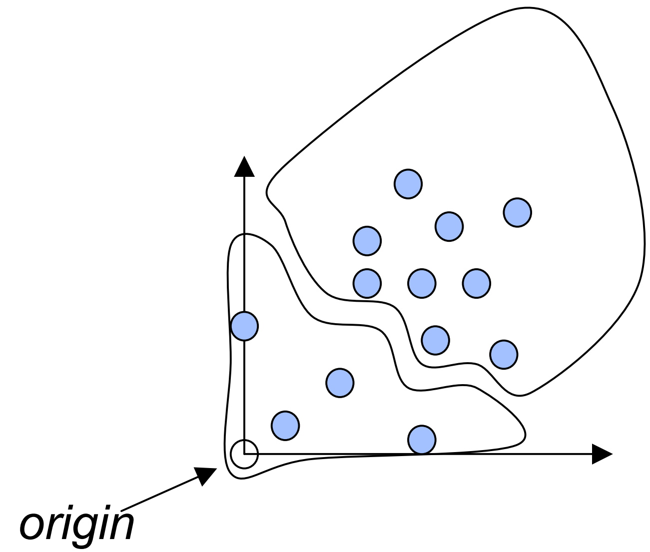
- **Use a replicator 4-layer feed-forward neural network (RNN) with the same number of input and output nodes**
- **Input variables are the output variables so that RNN forms a compressed model of the data during training**
- **A measure of outlyingness is the reconstruction error of individual data points.**



* S. Hawkins, et al. Outlier detection using replicator neural networks, DaWaK02 2002.

*Unsupervised Support Vector Machines for Outlier Detection**

- **Unsupervised SVMs attempt to separate the entire set of training data from the origin, i.e. to find a small region where most of the data lies and label data points in this region as one class**
- **Parameters**
  - **Expected number of outliers**
  - **Variance of rbf kernel**
    - **As the variance of the rbf kernel gets smaller, the separating surface gets more complex**



push the hyper plane away from origin as much as possible

* E. Eskin et al., A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data, 2002.

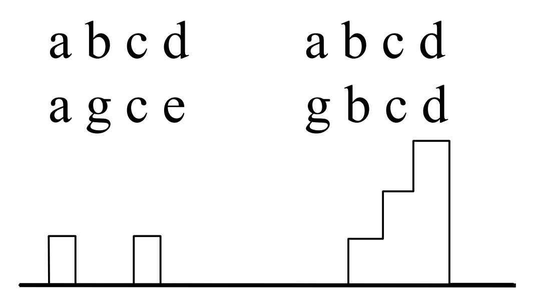
* A. Lazarevic, et al., A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection, SIAM 2003

Profiling based anomaly detection

- Profiling methods are usually applied to host based intrusion detection where users, programs, etc. are profiled
  - Profiling sequences of Unix shell command lines
  - Profiling users' behavior

Profiling: Temporal Sequence Learning*

- Data – sequences of Unix shell command lines
- Set of sequences (*user profiles*) reduced and filtered to reduce data set for analysis
- Build Instance Based Learning (IBL) model that stores historic examples of “normal” data
  - Compares new data stream
  - Distance measure that favors long temporal similar sequences
  - Event sequences are segmented



AHPCRC

* T. Lane, C. Brodley, Temporal Sequence Learning and Data Reduction for Anomaly detection, 1998.

*Profiling: Anomaly Detection using NNs **

- **Modeling the behavior of individual users**
- **Data – audit logs for each user for several days**
- **Form a distribution vector – how often user executes each command**
- **Train Neural Network with these vectors as inputs**
- **Identify whether the user is regular or illegal for each new command distribution vector, I.e for each new login session**

* J. Ryan, M. Lin: Intrusion Detection with Neural Networks, *NIPS*, 1998.

Profiling: NNs for Anomaly Detection *

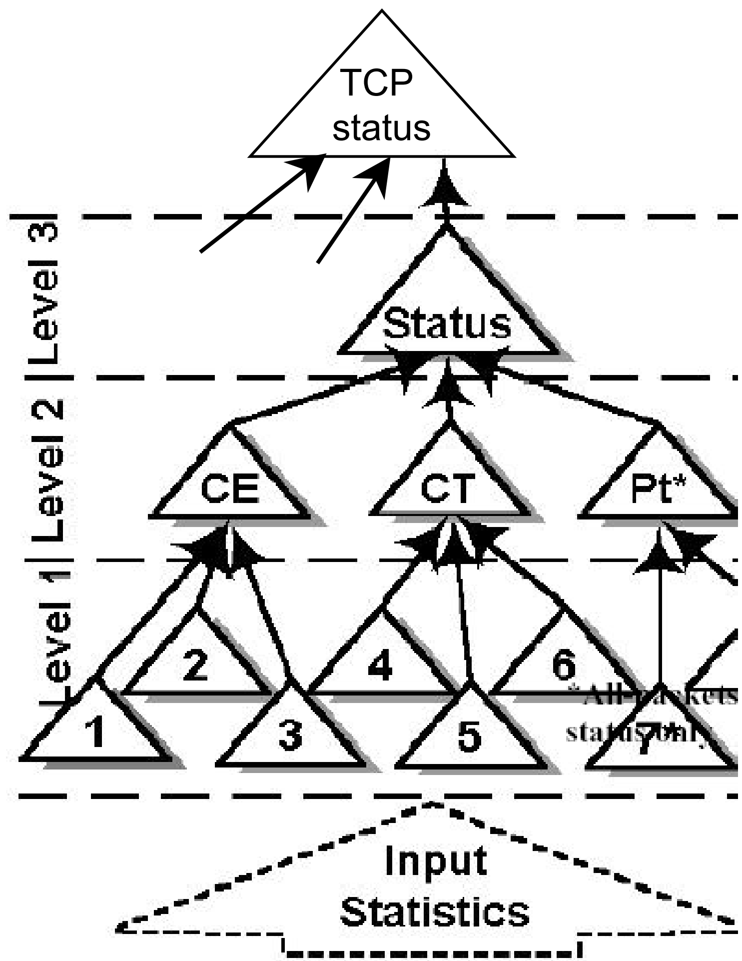
- Build profiles of software behavior and distinguish between normal and malicious software
- Data – strings of BSM (Basic Security Module) events
- Classify entire sessions not single strings of BSM events
- NN with one output node
  - "leaky" bucket algorithm employed
  - leaky bucket algorithm keeps a memory of recent events by incrementing a counter of the neural network's output, while slowly leaking its value
  - If level in the bucket > threshold $\Rightarrow$ generate alarm
  - emphasizes temporal co-located anomalies

AHPCRC

* A. Ghosh, A. Schwartzbard, A Study in Using Neural Networks for Anomaly and Misuse Detection 1999.

*Profiling: NNs for Anomaly Detection **

- **Three-level architecture**
  - **Packets and queue statistics are used as inputs to the level 1 NNs**
  - **The outputs from the Level 1 NNS are combined into:**
    - **Connection establishment (CE)**
    - **Connection termination (CT)**
    - **Port use (*Pt* for all packets only)**
  - **Outputs from Level 2 are combined at Level 3 into a single status**
  - **Each of these status monitors are further combined to yield a single TCP status**



* S. Lee, D. Heinbuch, Training a Neural-Network based Intrusion Detector to Recognize Novel Attacks, 2000.

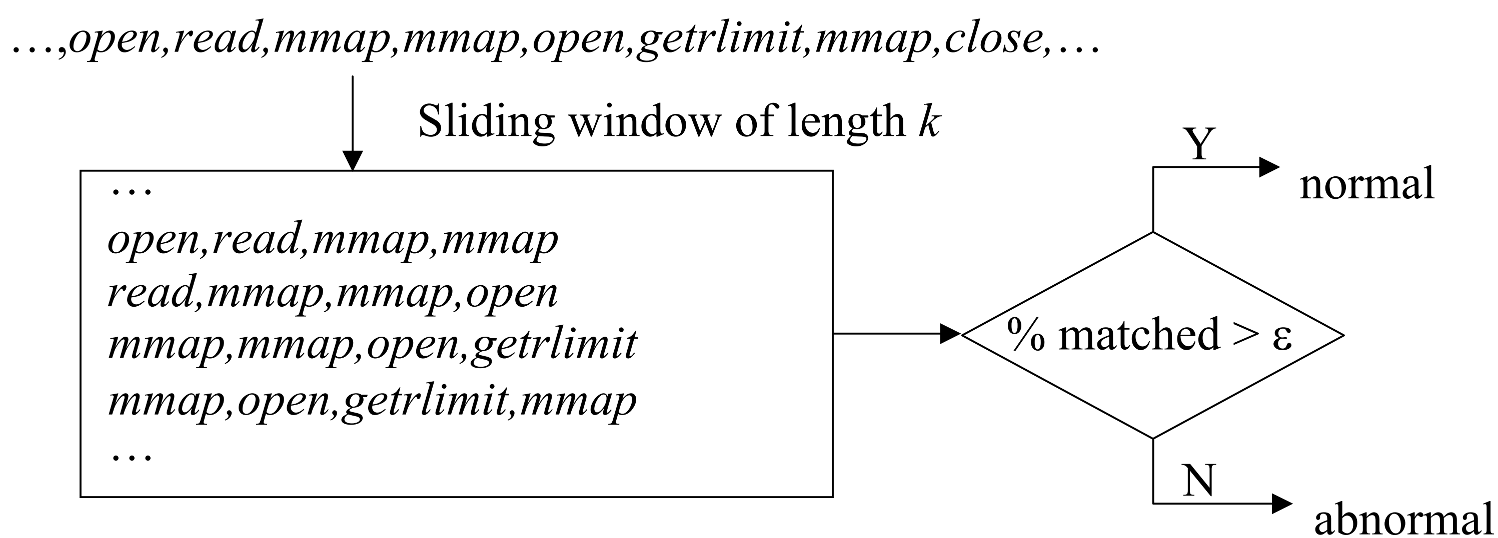
*Profiling: Data Mining Analysis of RTID Alarms**

- Data – stream of intrusion alarms generated by IBM Network Operations Center (NOC)
- Normal stream of alarms is modeled using association rules (AR) (*Intelligent Miner*)
  - Frequent itemsets
  - Association rules (AR) with high confidence
- Incoming new stream of alarms
  - Check for the known frequent itemset (Yes → Normal)
  - No → Unexpected absence of alarms, Apply AR

* S. Manganaris, et al., A Data Mining Analysis of RTID alarms, 1999.

Modeling System Calls Data*

- Learn program behavior profiles from previous execution (short sequences of system calls)
  - …open read mmap mmap open close … Unique sequences for window size 3:
- Learn only traces from system calls from normal data
  - Detect deviation from this profile[1]
- Learn traces from both normal and intrusive system calls
  - Train a RIPPER classifier that will learn classes[2]



1. S. Hofmeyr, et al, Intrusion Detection using Sequences of System Calls, 1997.
2. W. Lee, et al, Learning Patterns from Unix Process Execution Traces for Intrusion Detection, 1997.

Alternative Approaches

- *Artificial Anomalies Generation**
- **For sparse regions of data generate more artificial anomalies than for the dense data regions**
  - **For each attribute value *v* generate [(# of occurrence for most frequent value) – (# of occurrences for this value)] anomalies ($v_a \neq v$, other attributes random from data set)**

Values of Attribute *i*	Number of occurrences	Number of generated examples
A	1000	-
B	100	900
C	10	990

- **Filter artificial anomalies to avoid collision with known instance**
- **Use RIPPER to discover rule sets**
- **Pure anomaly detection vs. combined misuse and anomaly detection**

* W. Fan et al, Using Artificial Anomalies to Detect Unknown and Known Network Intrusions, IEEE ICDM 2001.

*Distributed IDS Characteristics**

- **Data analysis is performed on a number of locations proportional to the number of hosts that are monitored**

Characteristics	Centralized IDS	Distributed IDS
Run continually	relatively small # of components	harder–large # of components
Fault Tolerant – recovery	IDS- centrally stored – **easier** to recover	more difficult to store in recoverable & consistent manner
Resist subversion	small #, but large components	large # components, cross-check
Minimal Overhead	may be large for large loads	small – components smaller
Configurable	easier, small # of components	easy, component localized to hosts
Adaptable	few locations-easier to detect global changes, local behavior harder	more difficult for global changes local changes are easier to detect
Scalable	more computing and storage resource	large scale – adding components
Graceful serv. degradation	component stops working, IDS stops	component stops, IDS may persist
Dynamic reconfiguration	component $\Rightarrow$ Need to restart IDS	restart components-no affect on IDS

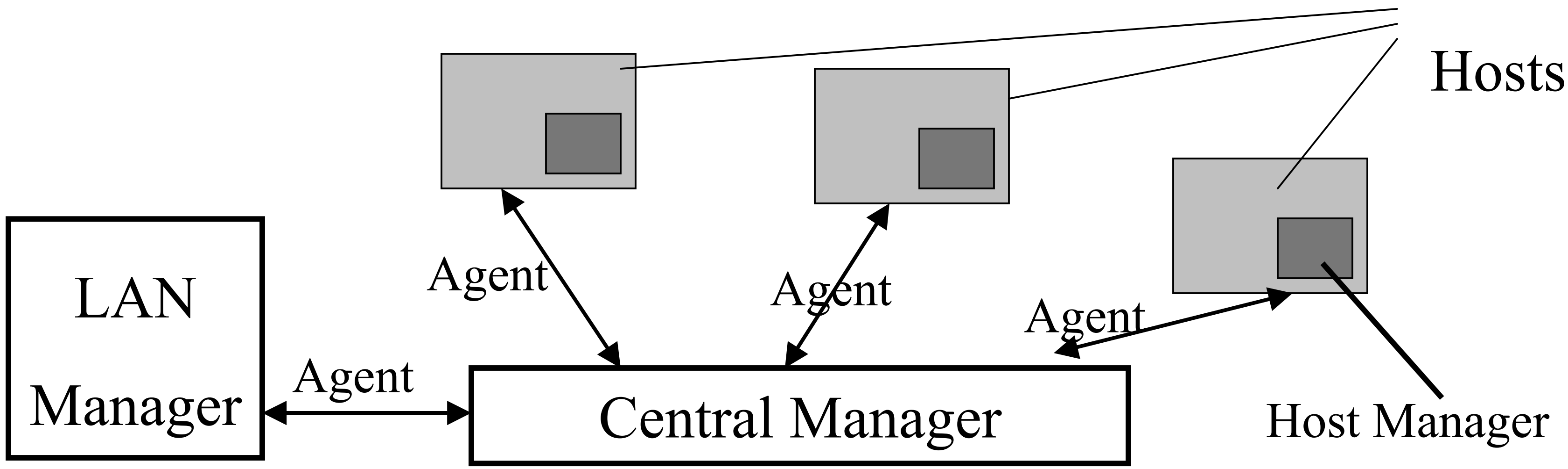
AHPCRC

* E. Spafford, D. Zamboni, Intrusion Detection using Autonomous Agents, *Computer Networks*, 2001.

Distributed IDS

- **A System for Distributed Intrusion Detection, S. Snapp, 1991.**
- **An Introduction to Distributed IDSs, N. Einwechter, 2001.**
- **A Distributed Autonomous-Agent NID and Response System, J. Barrus, N. Rowe, 1998.**
- **Intrusion Detection using Autonomous Agents, E. Spafford, D. Zamboni, *Computer Networks*, 2001.**
- **Intelligent Agents for Intrusion Detection, G. Helmer, 1999.**
- **A Large-Scale Distributed ID Framework …, M. Huang, 1999**

A System for Distributed Intrusion Detection*



- Host managers responsible for detecting single independent events, sequence of events (known attacks through pattern matching), anomalies (inform Central Manager)
- LAN Manager audits host-host connections, used services, traffic volume and analyzes unusual network activities
- Central Manager based on an expert system (E, IR, IM) – high level data analysis (correlation of data from hosts)

* S. Snapp, A System for Distributed Intrusion Detection, 1991.

DIDS

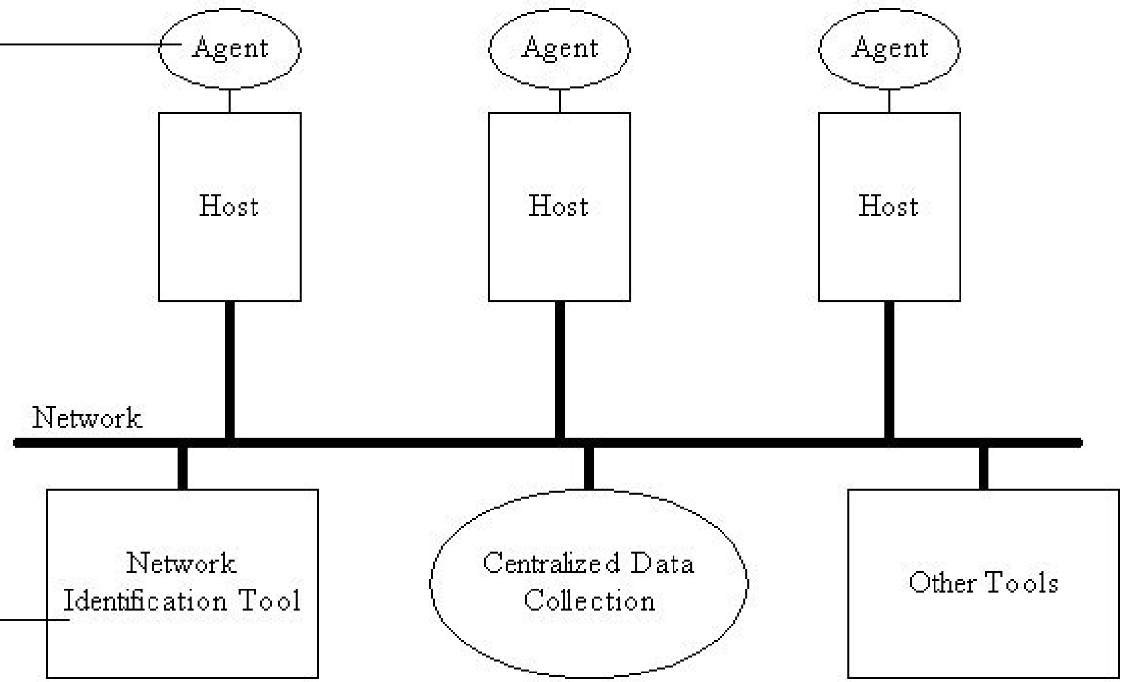
- **DIDS consists of multiple *IDS agents* over a large network that communicate with each other**
- **Central Analysis Server consists of database and Web server**
- **IDS Agents are collecting attack information world wide and send them to the central server**
- **Attack aggregation from agent network – e.g. aggregate attacks according to attacker IP, attacked port, …..**

AHPCRC

* N. Einwechter, An Introduction to Distributed IDSs, 2001.

Distributed Autonomous Agents*

Agents monitor hosts & communicating with each other



recognizes coordinated attacks distributed through network

- **Two main factors of *alert level* = *danger * transferability***
  - **Danger (5 levels: minimal, cautionary, noticeable, serious, catastrophic)**
  - **Transferability (3 levels: none (local environment), partial, full)**
- **3 alert levels: normal, partial alert, full alert**
- **Use neural networks with 8 features from statistics over time**

AHPCRC

* J. Barrus, N. Rowe, A Distributed Autonomous-Agent NID and Response System,, 1998.

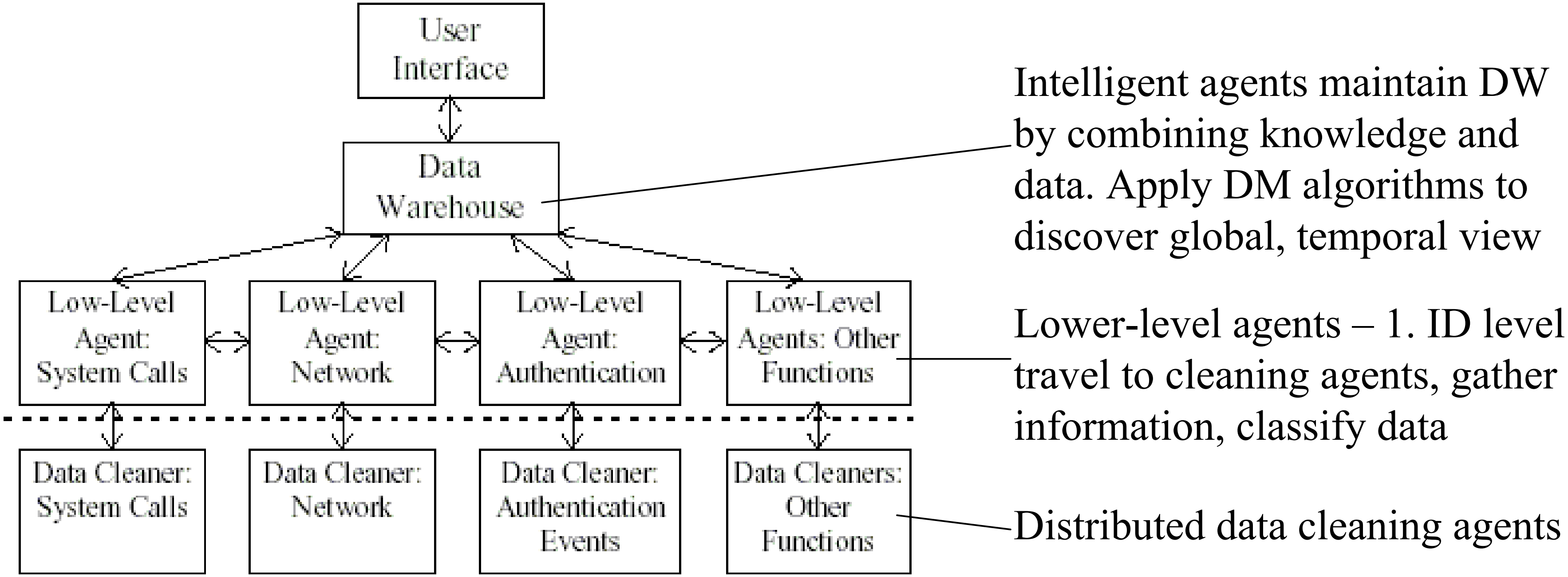
*AAFID - Autonomous Agents for ID**

- **AAFID components**
  - ***agents*** **monitor for interesting events, send messages to transceiver, may evolve over time using Genetic Programming (GP), may migrate from host to host**
  - ***filters*** **- data selection and data abstraction layer for agents that specify which records they need and what data format**
  - ***transceivers*** **– control (keeps track of agent execution) and data processing (process info from agents)**
  - ***monitors*** **– control and data processing from different hosts**
- **GP agents are trained on generated scenarios, where each agent is assigned a *fitness score* according to its accuracy**

AHPCRC

* E. Spafford, D. Zamboni, Intrusion Detection using Autonomous Agents, Computer Networks, 2001.

*Intelligent Agents for NID**



IDS Architecture

- **System call traces data set**
- **RIPPER – classification algroithm**

* G. Helmer, Intelligent Agents for Intrusion Detection, 1999.

Using Bayesian Methods in Distributed IDS*

- More effectively analyze information provided by existing IDSs from multiple networks
  - Intrusion events from multiple IDSs are collected
- Bayesian Multiple Hypothesis Tracking (BHMT):
  - Generate and store all possible hypotheses that explain the measured intrusion events
  - To determine the likelihood of the hypotheses, they are evaluated against the understanding of the sensor behavior
  - The hypothesis with the greatest likelihood is assumed correct
  - Each hypothesis consists of a set of tracks that map new events to existing target tracks, new target tracks or false alarms
  - As new intrusion event arrives, the likelihood of hypotheses is either strengthen or weaken (assumed a false alarm)
  - Drawback; large number of possible hypotheses that describe intrusion events

* D. Burroughs, Analysis of Distributed IDSs Using Bayesian Methods, 2001.

Benchmarking IDSs*

- **Measures:**
  - **TP and TN rate**
  - **reduced FP and FN rate (FP vs. FN)**
  - **cost of misclassifications**
  - **operational cost (IDS slows down the network)**
  - **Efficiency (how fast the attack is detected)**
- **Knowing *what not* to measure**

* M. Ranum, Experience Benchmarking Intrusion Detection Systems, NFR Security, December 2001.

Surveys of Intrusion Detection

- IDSs: A Survey and Taxonomy, S. Axelsson, 2000.
- *A Revised Taxonomy for IDSs, H. Debar, M. Dacier, A. Wespi, 1999, 2000*.
- Malicious and Accidental Fault Tolerance for Internet Applications, IBM Research Lab, Zurich, 2001.
- Computer System Intrusion Detection: A Survey, A. Jones, 2000
- Classification And Detection Of Computer Intrusions, S. Kumar, 1995.

Intrusion Detection Links

- http://www.cs.umn.edu/~aleks/intrusion_detection.html
- http://www.cc.gatech.edu/~wenke/ids-readings.html
- http://www.cerias.purdue.edu/coast/intrusion-detection/welcome.html
- http://www.cs.ucsb.edu/~rsg/STAT/links.html
- *http://cnscenter.future.co.kr/security/ids.html* !!!!!
- *http://www.cs.purdue.edu/homes/clifton/cs590m/* !!!!!
- http://dmoz.org/Computers/Security/Intrusion_Detection_Systems/
- http://www.networkice.com/Advice/Countermeasures/Intrusion_Detection/default.htm
- http://www.infosyssec.net/infosyssec/intdet1.htm

Questions?

Thank You!

Contact: aleks@cs.umn.edu

srivasta@cs.umn.edu

kumar@cs.umn.edu

AHPCRC