Pattern Recognition in Computer and Network Security

Claudio Mazzariello
claudio.mazzariello@unina.it

June 01, 2010

Outline

- Security issues
  - Security requirements
  - Causes of security issues
  - Security violation detection as a pattern recognition problem
- Practical examples
  - Intrusion detection
  - Traffic classification
  - Botnet detection
  - VoIP security
    - Social threat detection

Security requirements

- Confidentiality
  - Grant access to data only with the information owner's authorization (either implicit or explicit)
- Integrity
  - Prevent unauthorized users from modifying the system's status or any piece of data residing in, or flowing through, the system
- Availability
  - Granting access to information or system resources to an authorized user
- Control
  - Preventing unauthorized users from gaining access privileges violating the system's access control policies

Common causes for security issues

- Protocols were designed for providing connectivity
- Networks were designed with the idea of “implicit” security
  - Few initial users
  - Reciprocal trust among users
  - Limited relevance of security-related issues
- Security requirements only satisfied at communication's endpoints
  - Cryptography
  - Authentication
  - ...

Cyber security threat

- Intrusion
  - Any action intentionally aimed at compromising the confidentiality, integrity, availability and control requirements
- Intruder or Attacker
  - Individual exploiting sequences of malicious actions with the aim of compromising systems or data

System Vulnerabilities

- Vulnerability
  - A “weakness” in a system allowing the execution of unauthorized actions
  - 7 new vulnerabilities are discovered every day
- Types of vulnerability
  - Software vulnerabilities
    - Caused by design errors
    - Caused by implementation errors
  - Configuration vulnerabilities
    - Production systems using default configuration parameters
    - Wrong choices of configuration parameters

System Vulnerabilities

- Present in each layer of the TCP/IP stack
  - Evidences show attacks targeting each of the layers
  - Attack evidences can be sought for in each of the layers
- Application-layer attacks
- Transport-layer attacks
  - TCP vulnerabilities
  - UDP vulnerabilities
  - …
- Network-layer attacks
  - IP-layer vulnerabilities
  - Routing protocol attacks
  - …
- ...

Attack victim

- Target:
  - Objective targeted by the malicious actions
- Potential attack targets
  - User accounts
  - Processes
  - Data
  - Hardware components
  - Host computer
  - Network
  - ...

Evolution of a Cyber Attack

- Reconnaissance
  - The attacker collects information about the potential victims, by searching for exposed vulnerabilities
- Vulnerability exploitation
  - The attack is performed
- Privilege acquisition
- Exploitation of acquired privileges for further malicious actions

Security countermeasures

- We have shown that attacks
  - Evolve in time through several phases
  - Can affect networked systems at different layers

- Structured defense strategies
  - Observation at different layers
  - Search for activities typical of each attack phase

Why Pattern recognition?

Signature-based approach is the most commonly used for Network Intrusion Detection

- The IDS has a set of manually written rules called attack signatures
- The content of network packets is inspected and an alarm is raised if it matches any of the rules
- Usually effective against known attacks + low FP
- Efficient algorithms for pattern/string matching

Problems related to signature-based IDS

- Signatures must be immediately updated for each new attack
- Manual signature generation is too slow
- Polymorphism makes signature-based IDS ineffective

Why pattern recognition?

Manual analysis of network data to extract classification rules is difficult/slow/expensive

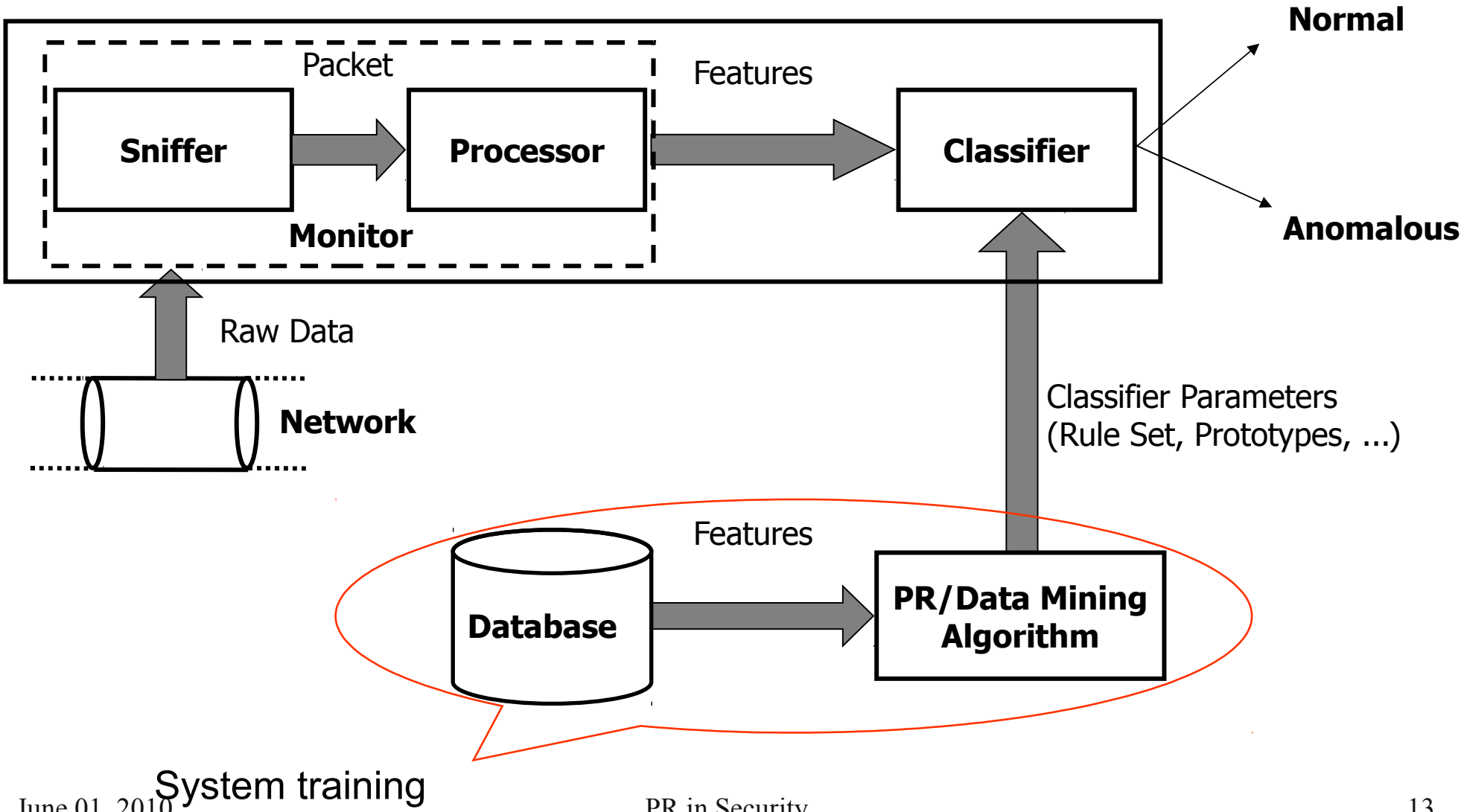
- Need reverse engineering observed attack
- Rules are usually exploit-based, instead of vulnerability based

We need an automatic way to create/update NIDS

We need higher generalization ability compared to manually generated signatures

- Detection of variants of known attacks (polymorphism)
- Detection of zero-day attacks

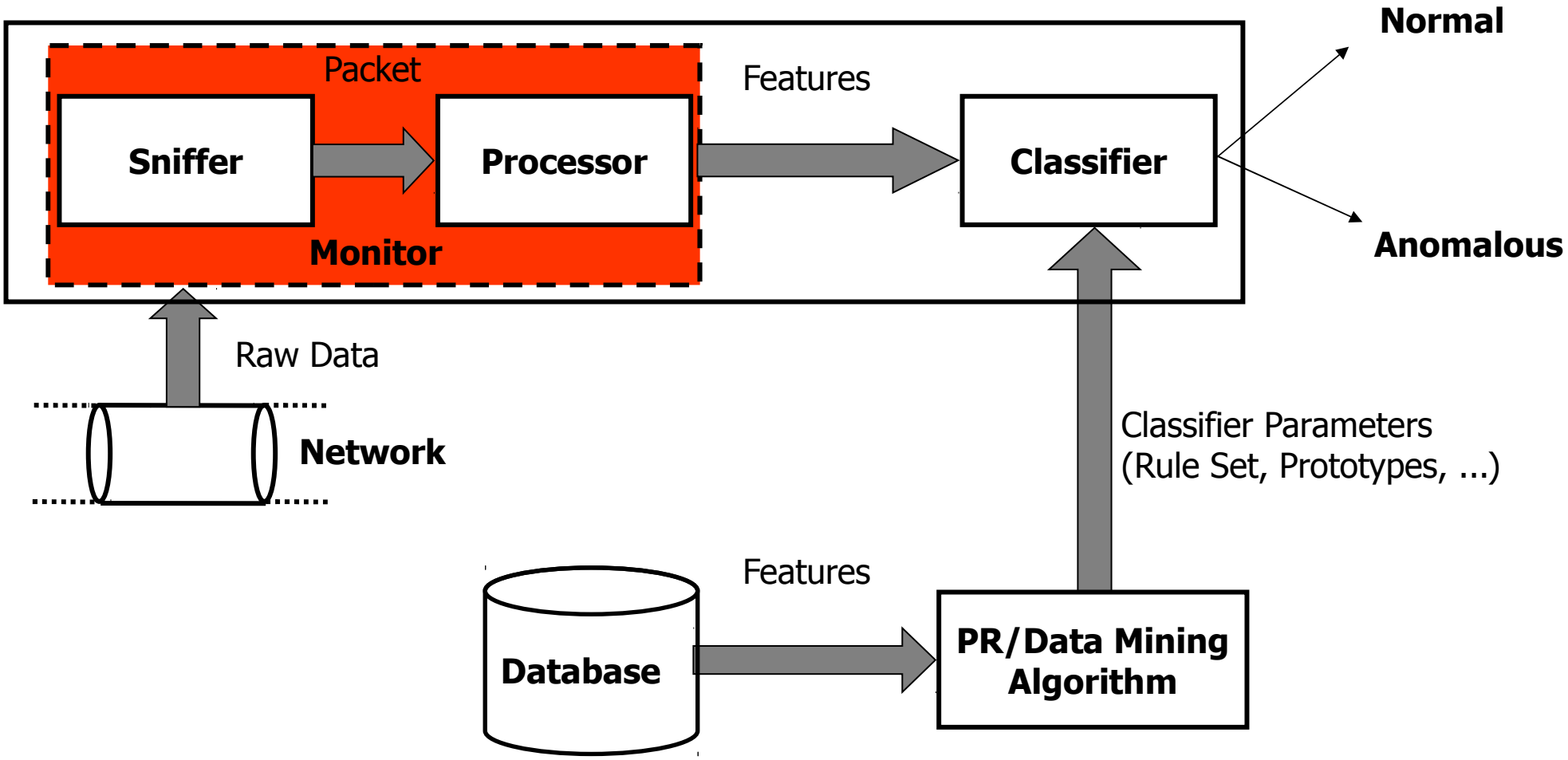
Reference Model



Design choices

- How do we represent traffic in a tractable manner
- How do we separate wanted and unwanted traffic
  - How does the system “learn” to decide
- How do we evaluate the system's accuracy

Traffic representation



Detecting Unwanted Activity with Pattern Recognition

- Pattern Recognition
  - Identify patterns with specific properties in data sources

- Pattern <==> Network user activity
  - Identify network user activity (patterns) with specific properties in network traffic (data source)

Pattern <==> Network user activity

- Define a suitable model for the problem at hand
  - Identify the problem
  - Identify meaningful data properties
- Real-time vs. non real-time
- Stateless vs. stateful

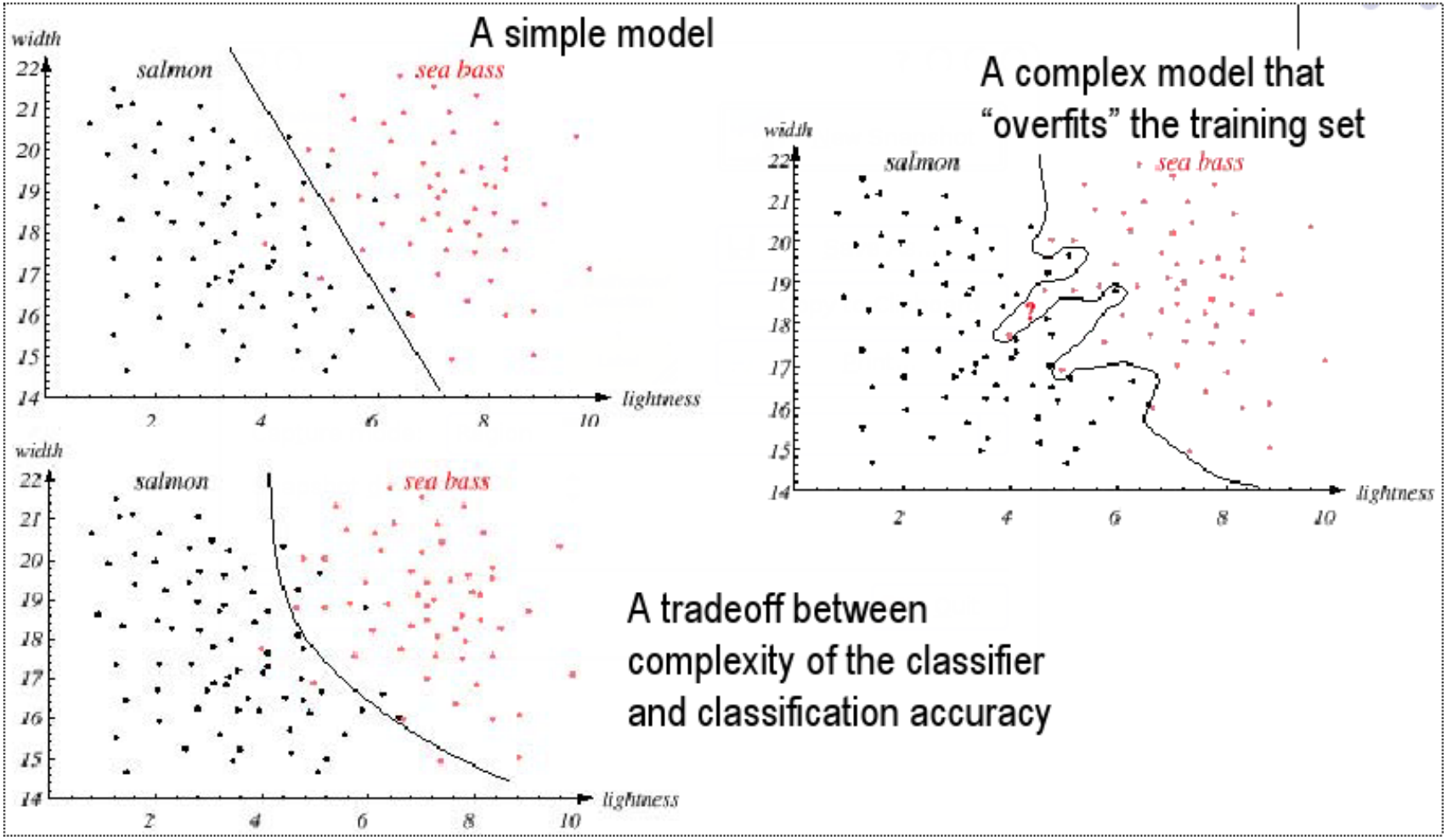
Defining a Model

- Describing network events is an “Art”
  - Depends on designer's intuition and expertise
  - Depends on the application scenario
- Different features for different services and classes of attacks we want to detect.
  - Application layer features
  - Transport layer features
  - Network layer features
  - …
  - Cross layer features

Defining a Model

- A pattern can be represented as a point in the features space
- Classification is then formulated as the task of finding the optimal separating surface between normal activities and intrusions
  - Optimal in the sense of error minimisation
- The estimation of the separating surface requires a training set of examples
  - The more representative the training set, the more effective the detection
- An independent test set is used to estimate the performance on new patterns

Decision Surfaces



Real-time vs. non real-time

- Real-time feature computation
  - Allows early detection
  - Allows early reaction and remediation
  - Must be QUICK!
    - Lower accuracy tolerated
- Non real-time feature computation allows the definition of more accurate features
  - Allows greater accuracy
  - Allows the definition of significant features
  - Results are not related to what is really going on in the network

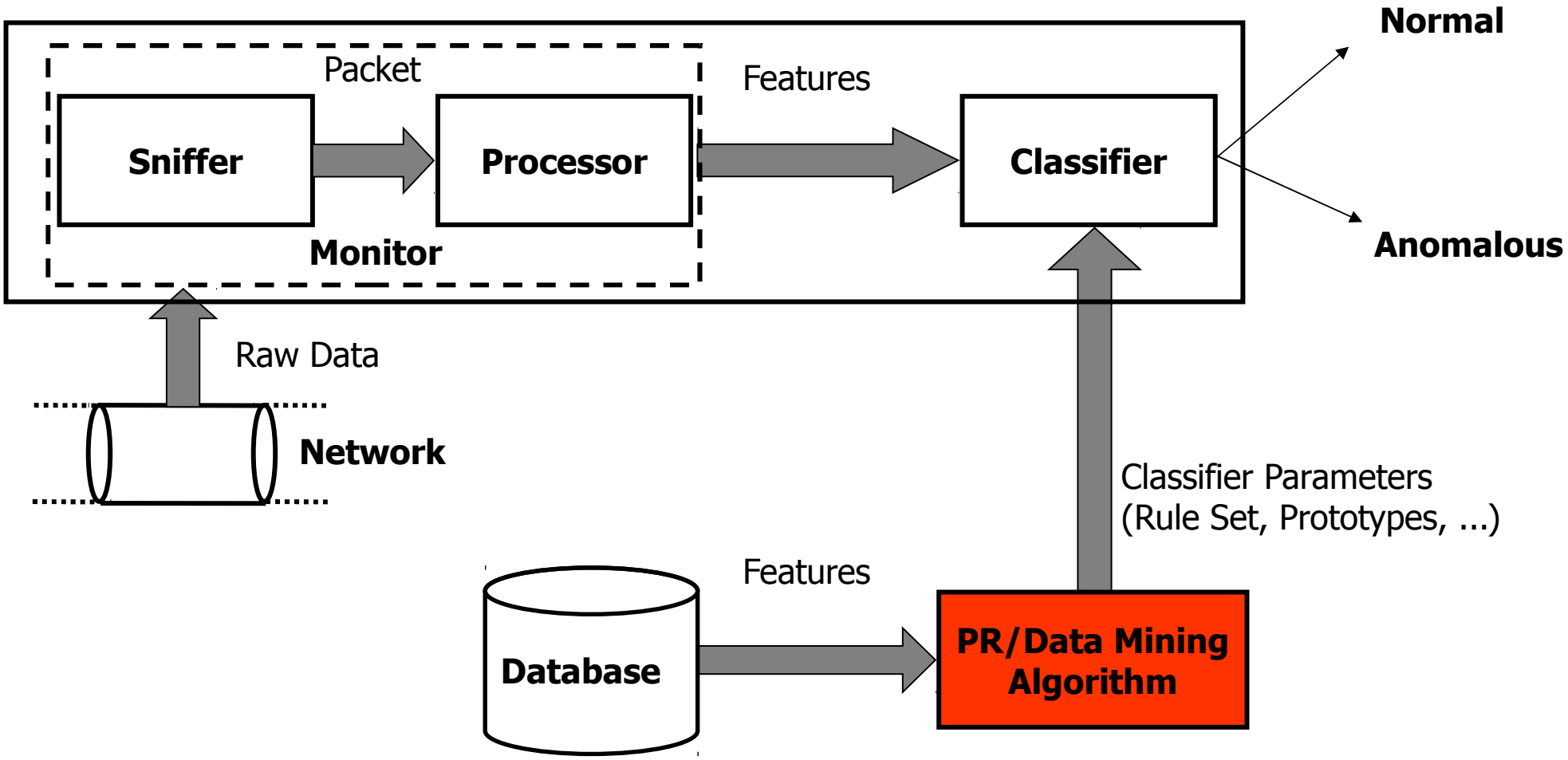
Stateless vs. Stateful Approaches

- Stateless approaches treat each event independently of others
  - Simple system design
  - High processing speed
- Stateful approaches maintain information about past events
  - The effect of a certain event depends on its position in the event stream
  - Complex system design compared to stateless approaches

Selection vs. Extraction

- Extraction methods: pros
  - The extraction method projects the original feature space onto a subspace chosen in order to preserve the maximum possible amount of information. It then presents an high flexibility (selection can be seen as a particular case of extraction).
- Selection methods: pros
  - Features given by a selection method are a subset of the original ones and then their physical meaning is preserved. This can be important when information about the interpretation of the various feature must be integrated during the classification process (e.g.: knowledge-based methods). On the contrary, the extraction method generates “virtual” features, defined by means of linear combinations of the original ones and then without a real physical meaning.

System Training

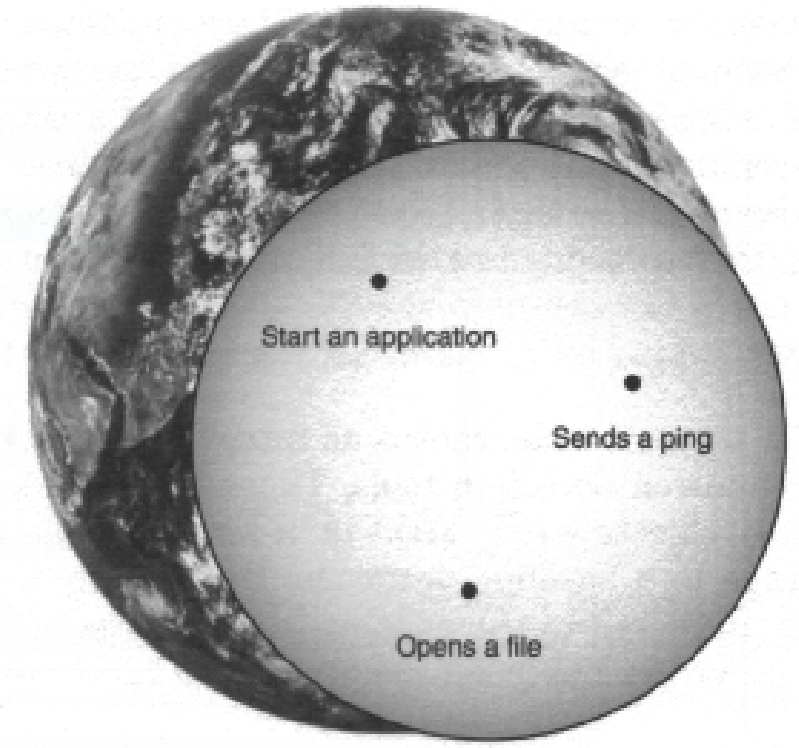


Misuse- vs. Anomaly-based

- Misuse-based
  - Patterns of known misuses are stored in a set of rules
  - When a rule is matched, an alarm is raised
  - Very good in detecting copycats attacks
- Anomaly-based
  - Statistical description of “normal” computer activities
  - All activities deviating from the normal profile are labelled as being anomalous
    - Not all of them are “unwanted”
  - Can detect “zero-day” attacks
  - Tends to produce high rates of false alarms

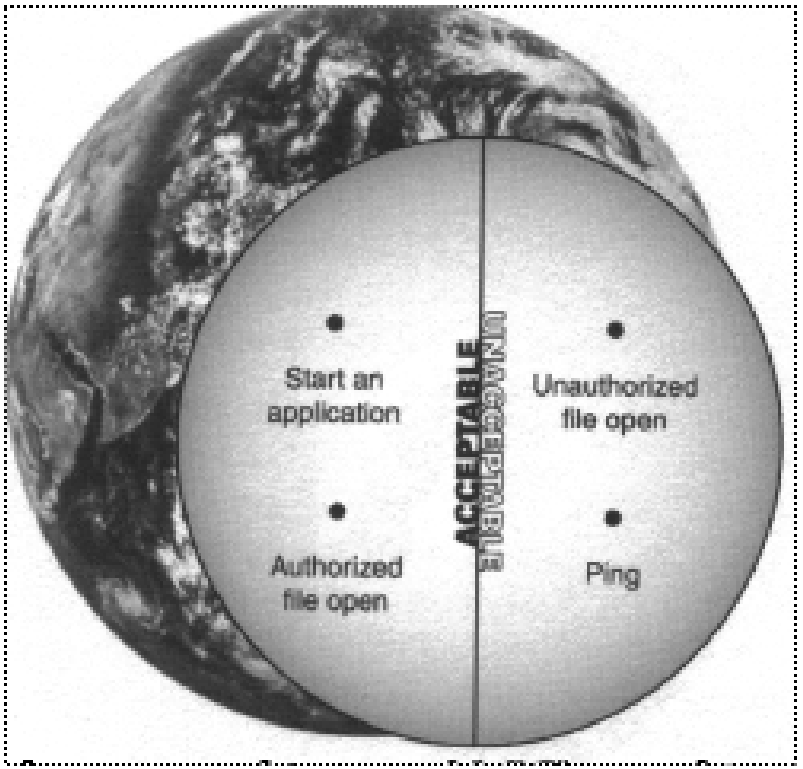
A conceptual model

- Proctor, Practical Intrusion Detection, 2001
- Approaches to Intrusion Detection can be described conceptually
- Let us represent a 2-dimensional feature space defined by a circle, representing all possible types of user behaviour and actions



A conceptual model

- We would like to define a feature space such that we can draw a line separating acceptable behaviour from unacceptable behaviour



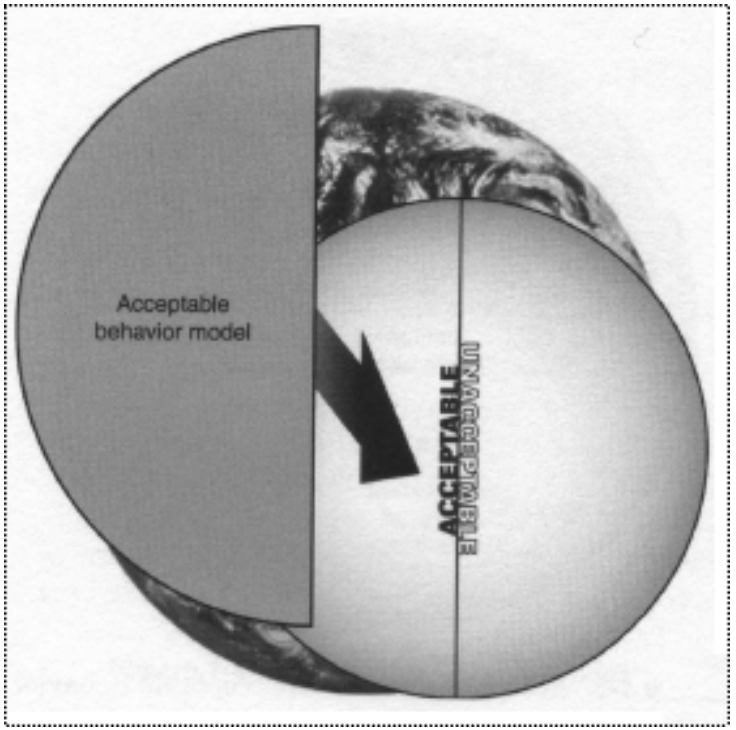
- Unfortunately, it is quite difficult to define such a feature space…

Anomaly detection

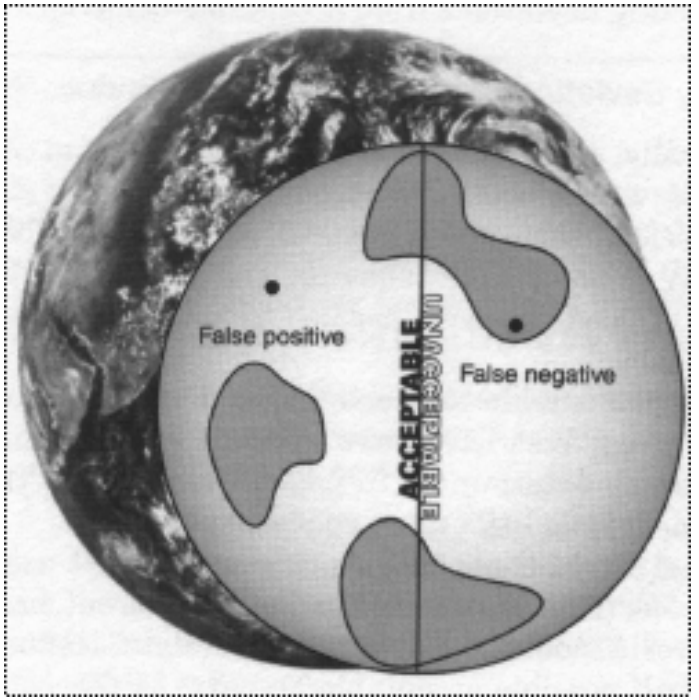
- If we could define everything that was acceptable, then everything that wasn't acceptable would be misuse
  - Historical data are used to define acceptable
- Unfortunately, it is very difficult to represent all possible acceptable activities
- When an acceptable action arises that has not been seen before, an alarm will be raised
- Additionally, unacceptable actions may exists in historical data so that unacceptable actions are considered acceptable

Anomaly detection, conceptually

Ideally…



Real-World Behaviour Models

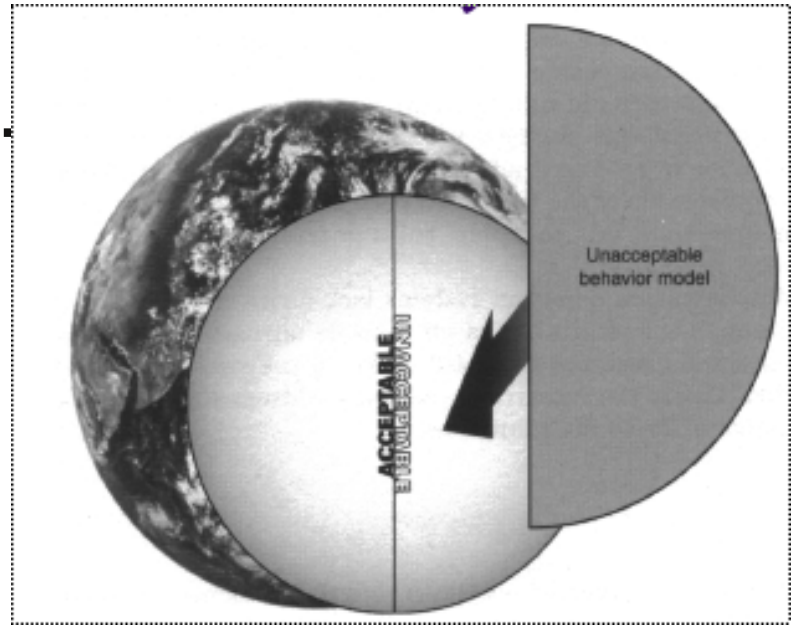


Misuse detection

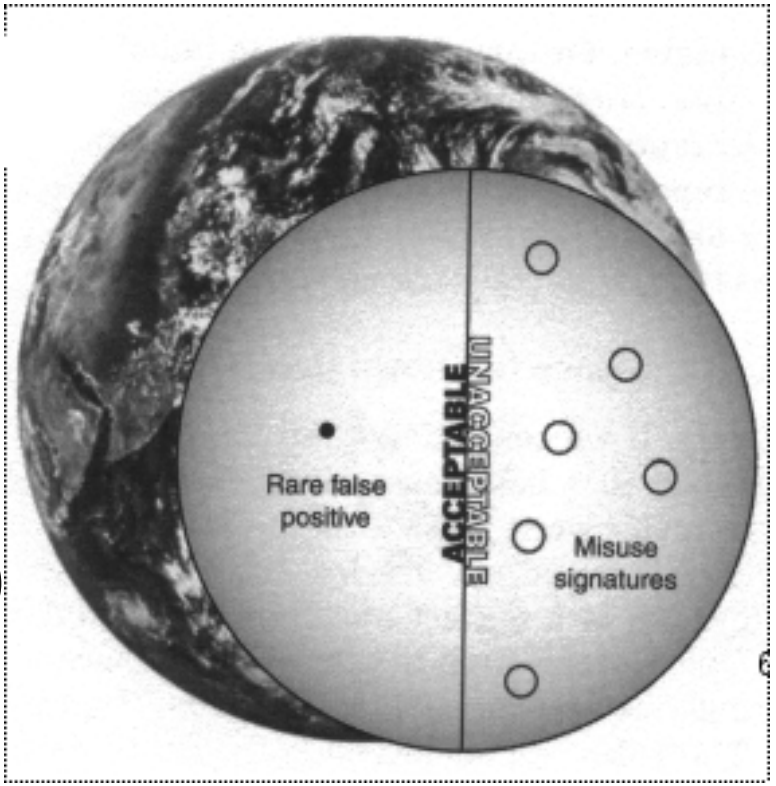
- Conversely, if we could define all unacceptable behaviour, then everything that matched that behaviour would result in an alarm
  - A priori rule-based mechanisms
- These predefined threat scenarios are very deterministic and significantly reduce the number of false-positives
  - There can be numerous missed alarms!
    - To reduce missed alarms, rules must be updated frequently with the most recent observed threats
- Actual alarms are very robust because they are focused on misuse activities

Misuse detection, conceptually

Ideally.



Real-World Models (Signatures)



Anomaly- or Misuse-based IDS?

The choice is governed by the trade-off between detection accuracy and false alarm rate

$$P(I|A) = \frac{P(I)\cdot P(A|I)}{P(I)\cdot P(A|I) + P(\neg I)\cdot P(A|\neg I)}$$

$$P(\neg I|\neg A) = \frac{P(\neg I)\cdot P(\neg A|\neg I)}{P(\neg I)\cdot P(\neg A|\neg I) + P(I)\cdot P(\neg A|I)}$$

Supervised vs. Unsupervised

- Supervised techniques
  - Need a labelled training set
  - Can be biased by the presence of outliers in the training set
  - Training “quality” can be evaluated
  - Need to be updated regularly
- Unsupervised techniques
  - Don't need previously labelled data
  - Can be easily updated
  - Training “quality” can't be evaluated

Supervised Misuse Detection

Problem formulation:

- Given a labeled dataset of network traffic that contains both normal and attack events, construct a classifier that is able to distinguish between normal and attack traffic, and also between different classes of attacks

Supervised Misuse Detection

- Problems related to the supervised approach
  - It is difficult and expensive to collect a labeled dataset of real traffic that contains both normal and attack traffic
  - Simulated traffic is not representative of a real network's traffic
    - Different networks offer different services and receive/send different traffic

Unsupervised Anomaly Detection

Problem formulation

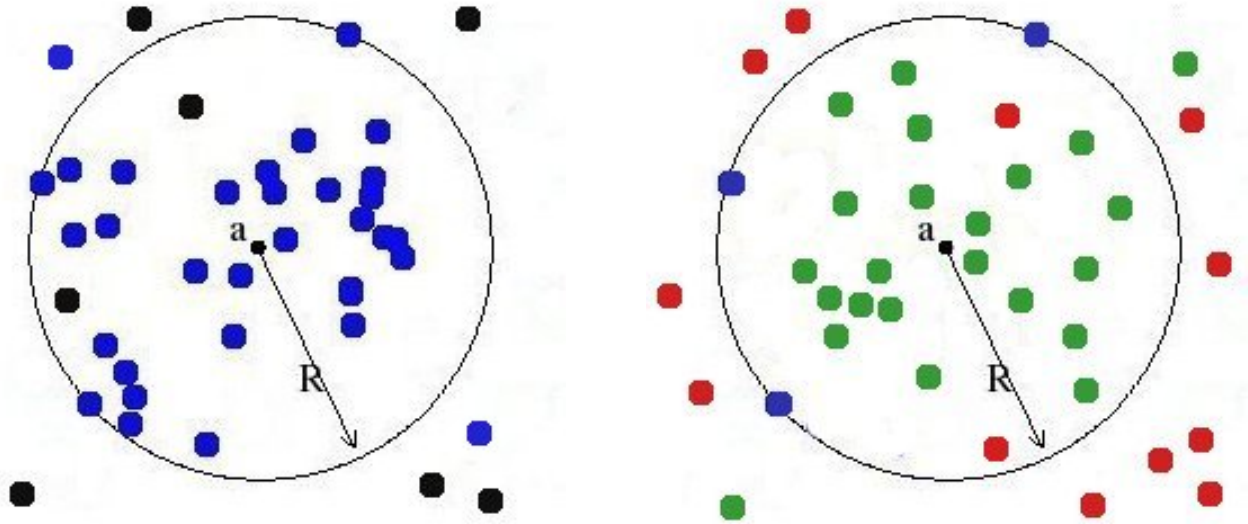
- Given a dataset of unlabeled network traffic, construct a classifier that can correctly recognize normal traffic as innocuous and flag attacks as anomalous events

Unsupervised Anomaly Detection

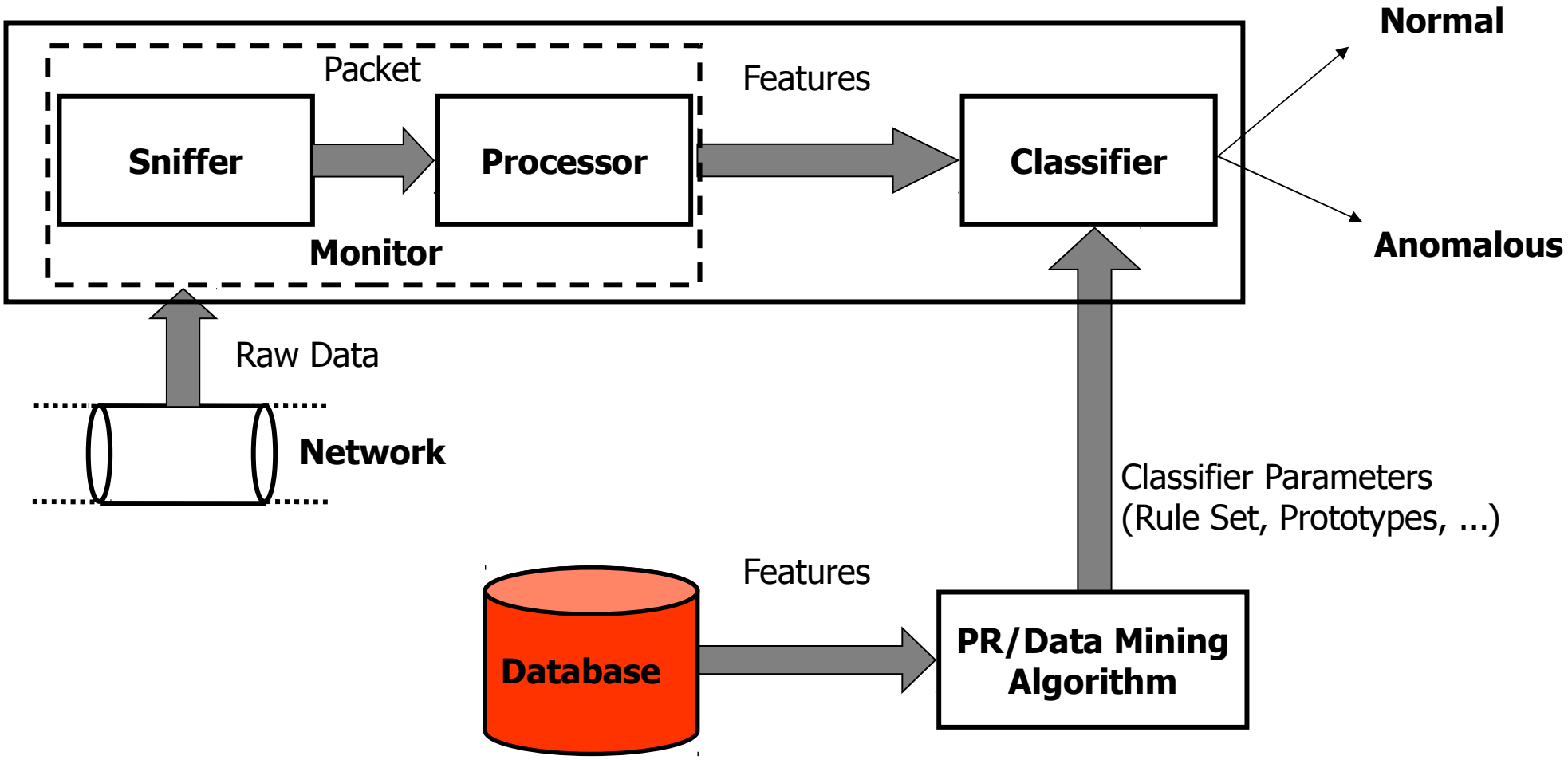
- It is really easy to sniff and store traffic live from a network
- Assumptions
  - The percentage of normal traffic is usually much higher than the percentage of anomalous traffic
  - We can filter known attacks
  - Using suitable metrics anomalous traffic can be separated from normal traffic
  - Unwanted behavior detection can be formulated as an Outlier-Detection or One-Class classification problem
- Advantage: detection of ZERO-DAY attacks

One-Class Classification

- Objective: distinguish between “target” objects and anything else
- Useful in case of highly unbalanced classes
- Can be used in case of unlabeled (noisy) dataset of “target” data



Reference data collection



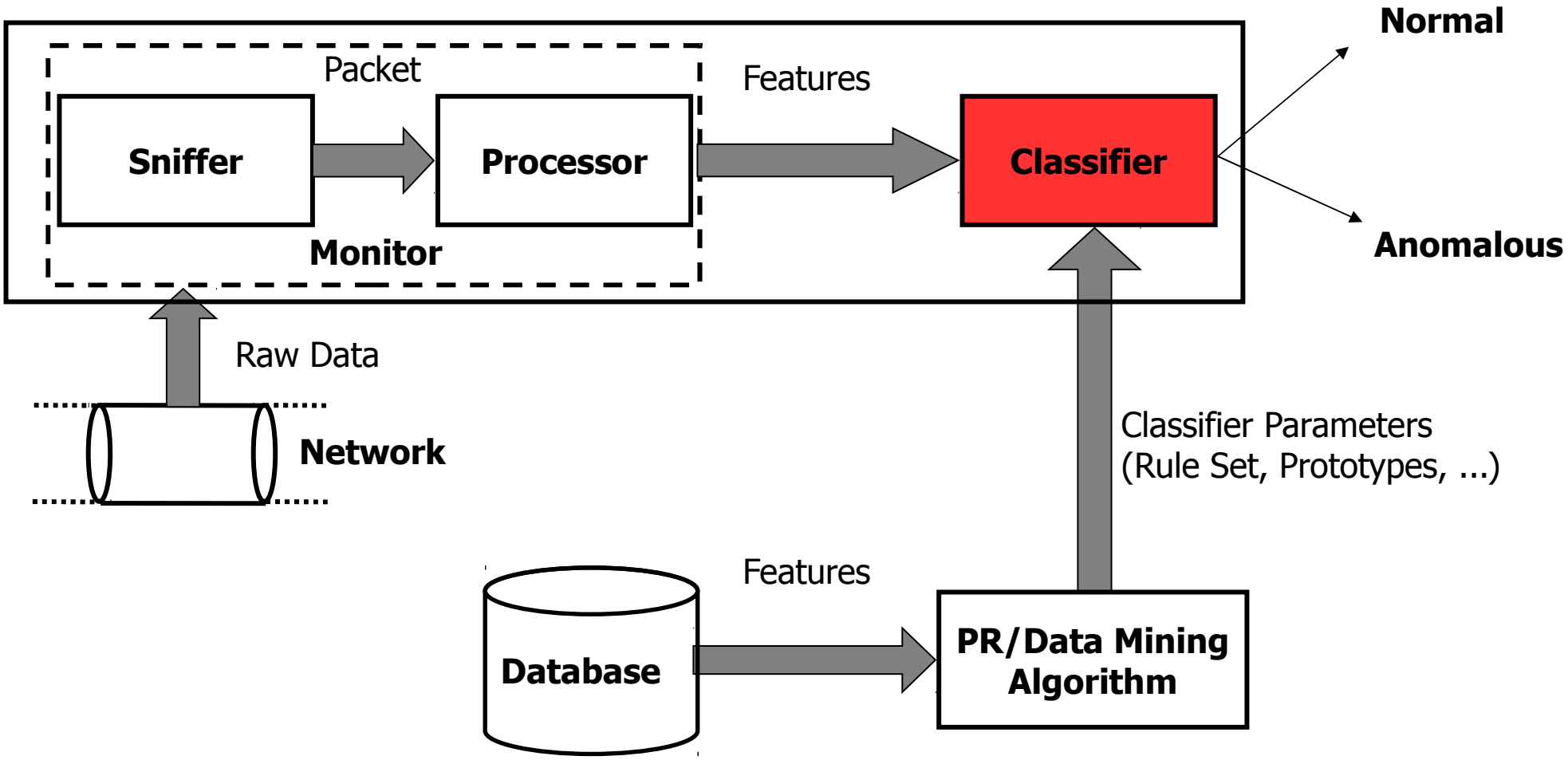
Data Collection

- Simulation
  - Pros & cons
- Emulation
  - Pros & cons
- Live capture
  - Pros & cons

User privacy

- Network data may contain sensible information
  - Telco operators NEVER give their data for analysis
- It's technically easy to sniff data
  - Sniffing private data can be prohibited
- Data must be anonymized
  - Header anonymization
  - Payload anonymization

Performance evaluation



Security is a complex system

- Security is a complex system that interacts with
  - itself
  - the assets being protected
  - the surrounding environment
- These interactions may cause failures even in the absence of attackers.
- These failures should be carefully examined as attackers are rarer than legitimate users.

Systems and how they fail

- Because security systems are designed to prevent attack, how the systems fail is critical
  - Active failures the system fails by taking action when it shouldn't
  - Passive failures The system fails to take action when it should

Attackers are generally rarer than legitimate users…

…how the systems fail in the absence of attackers (active failures) is generally more important than how the system fail in the presence of attackers (passive failures)

Active failures

- The system signals a false alarm (false positive)
  - The consequences can be merely irritating but also horrific, too
- Detection systems frequently suffer from rarity based failures
  - Trade-off between a high rate of false alarms or a significant number of missed alarms

Base-rate fallacy

- Axelsson (ACM Trans. Information and System Security, 2000) pointed out that the false alarm rate is the limiting factor for the performance of an intrusion detection system
- He used Bayes theory to show the trade-offs involved in designing an intrusion detection system.

Base-rate fallacy

- Let I and ¬I denote intrusive and nonintrusive behaviour respectively
- Let A and ¬A denote the presence or absence of an intrusion alarm
- Detection rate: P(A|I) (estimated by tests)
- False Alarm Rate: P(A|¬I) (estimated by tests)
- False Negative Rate: P(¬A|I) (estimated by tests)
- True Negative Rate: P(¬A|¬I) (estimated by tests)

  - P(¬A|I) = 1 - P(A|I) P(¬A|¬I) = 1 - P(A|¬I)

Base-rate fallacy

- For intrusion detection to be effective, both
  - P(I|A) (an alarm really indicates an intrusion)
  - P(¬I|¬A) (no alarm signifies no intrusion)
    - should be as large as possible
- From Bayes theorem

$$P(I \mid A) = \frac{P(I) \cdot P(A \mid I)}{P(I) \cdot P(A \mid I) + P(\neg I) \cdot P(A \mid \neg I)}$$

$$P(\neg I \mid \neg A) = \frac{P(\neg I) \cdot P(\neg A \mid \neg I)}{P(\neg I) \cdot P(\neg A \mid \neg I) + P(\neg I) \cdot P(\neg A \mid I)}$$

Base-rate fallacy - An example

- Suppose we have
  - 10 audit records per intrusion
  - 2 intrusions per day
  - 1.000.000 audit records per day
- Then

$$P(I) = \frac{1}{1\,e^{6}\,2 \cdot 1\,0} = 2\,e^{-5}$$

$$P(\neg I) = 1 - P(I) = 0{,}99998$$

$$P(I \mid A) = \frac{2e^{-5} \cdot P(A \mid I)}{2e^{-5} \cdot P(A \mid I) + 0{,}99998 \cdot P(A \mid \neg I)}$$

Base-rate fallacy - An example

- If $P(A|I) = 1.0$
  and $P(A|\neg I) = 10^{-5}$
  then $P(I|A) = 0.66$

- If $P(A|I) = 0.7$
  and $P(A|\neg I) = 10^{-5}$
  then $P(I|A) = 0.58$

- that is, half of the alarms are not caused by intrusions!

- If the false alarm rate is not as low as supposed, figures can even get worse!

Base-rate fallacy – An example

Analogously

$$P(\neg I \mid \neg A) = \frac{0.99998 \cdot (1 - P(A \mid \neg I))}{0.99998 \cdot (1 - P(A \mid \neg I)) + 2e^{-5} \cdot (P(A \mid I))}$$

that is, we will set off the alarm too many times in response to non-intrusions, combined with the fact that... we don't have many intrusions!

Types of errors

- Missed detections
- False alarms
- What about reject?
  - Detection accuracy evaluation
  - Reject unaccurate decisions

Intrusion Detection

Definitions

- Intrusions
  - Unauthorised access to, and/or activity in, an information system (IDSG, 1997)
  - Attacks originating outside the organisation (ICSA-IDSC, 1999)
- Intrusion Detection Systems (IDS)
  - Systems able at identifying that an intrusion has been attempted, is occurring, or has occurred (IDSG, 1997)
  - Systems that collect information from a variety of system and network sources, and then analyse the information for signs of intrusion and misuse (ICSA-IDSC, 1999)

Assets

- Host-based IDS
  - Aimed at detecting attacks related to a specific host
  - Tailored to a particular architecture/operating system
  - Detection is based on processing high level information (system calls, events, etc.)
- Network-based
  - Aimed at detecting attacks towards hosts connected to a LAN
  - Detection is based on processing data at lower level of granularity (packets)
- Common features
  - Analysis of discrete time-sequenced events

Host-based IDS

- Many host data sources
  - Operating systems event logs (kernel, BSM security, etc.)
  - Application logs (syslog, relational databases, web servers, etc.)
- Effective in detecting insider misuse
- Expensive, as host-based IDSs are typically distributed agent-based architectures

Network-based IDS

- Network sources are unique
- Network packets are usually sniffed off the network
- Sensors deployed throughout a network
- Most network-based attacks are directed at vulnerabilities of the operating system or application software

Host- and Network- based benefits

Benefit	Host	Network
Deterrence	Strong deterrence for insiders	Weak deterrence for insider
Detection	Strong insider detection Weak outsider detection	Strong outsider detection Weak insider detection
Response	Weak real-time response Good for long-term attacks	Strong response against outsider attacks
Damage Assessment	Excellent for determining extent of compromise	Very weak damage assessment capabilities
Attack anticipation	Good at trending and detecting suspicious behavior patterns	None
Prosecution support	Strong Prosecution support capabilities	Very weak because there is no data source integrity

Host- and Network- based benefits

- Network-based benefits
  - Cost of ownership
  - Packet analysis
  - Evidence removal
  - Malicious intent detection
  - Operating System independence
- Host-based benefits
  - Attack verification
  - System specific activity
  - Encrypted and switched environments
  - Monitoring key components
  - No additional hardware

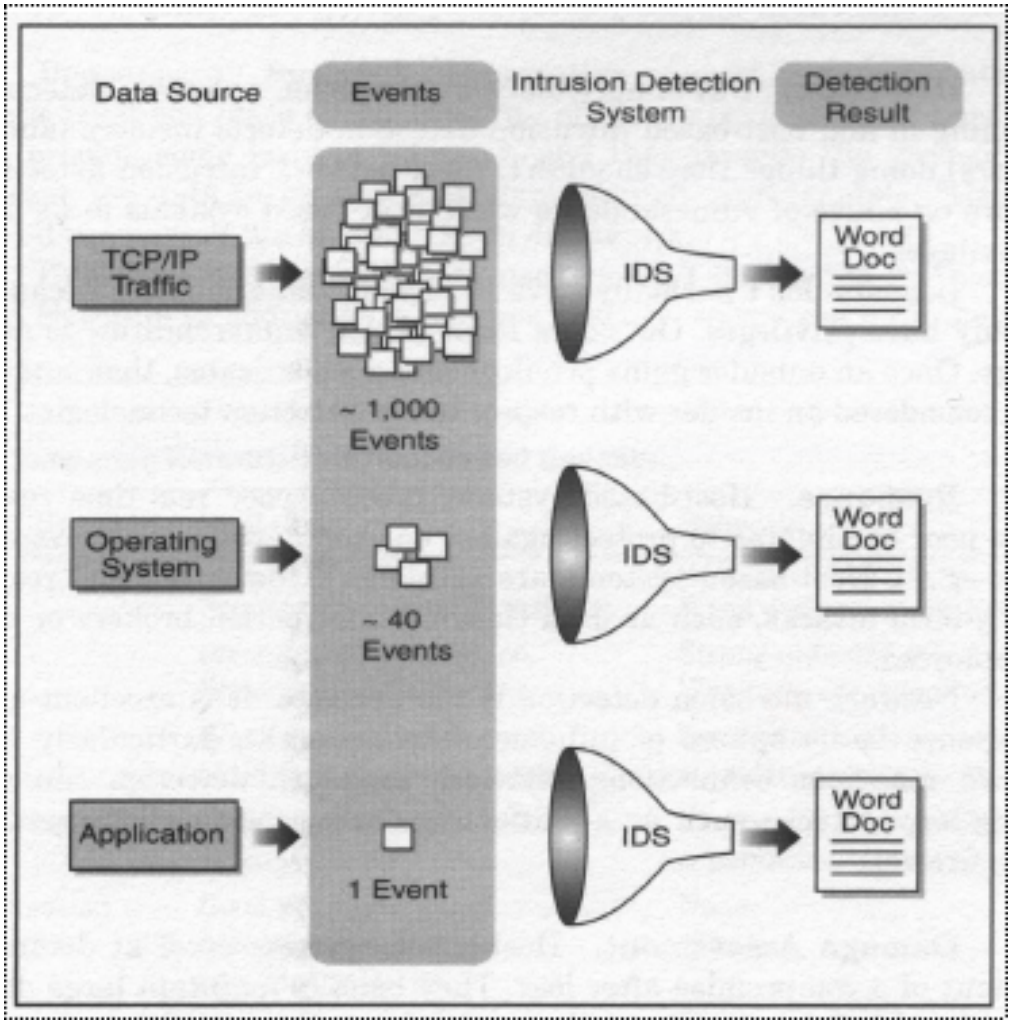
Host- or Network- based?

- Today emphasis is on network IDS
  - Attacks are performed through the Internet
  - Network IDSs allow for perimeter defence
  - Network IDSs not only detect attacks that exploit vulnerabilities in the communication protocol, but also vulnerabilities of operating systems and applications
  - Last but not least… network IDSs are appliances sold by those who also sell network appliances

… however, remember that IDS should be thought of as a component of a security strategy!

Host- vs. Network-based detection

Proctor, Practical Intrusion Detection, 2001



Traffic representation 1/2

- Parameters associated to network packets may be referred to:
  - A single packet
  - The connection which the packet belongs to
  - Statistical analysis of the relations between packets and connections sharing common properties

Traffic representation 2/2

We adopted a model inspired to the one proposed by W. Lee & S. J. Stolfo*

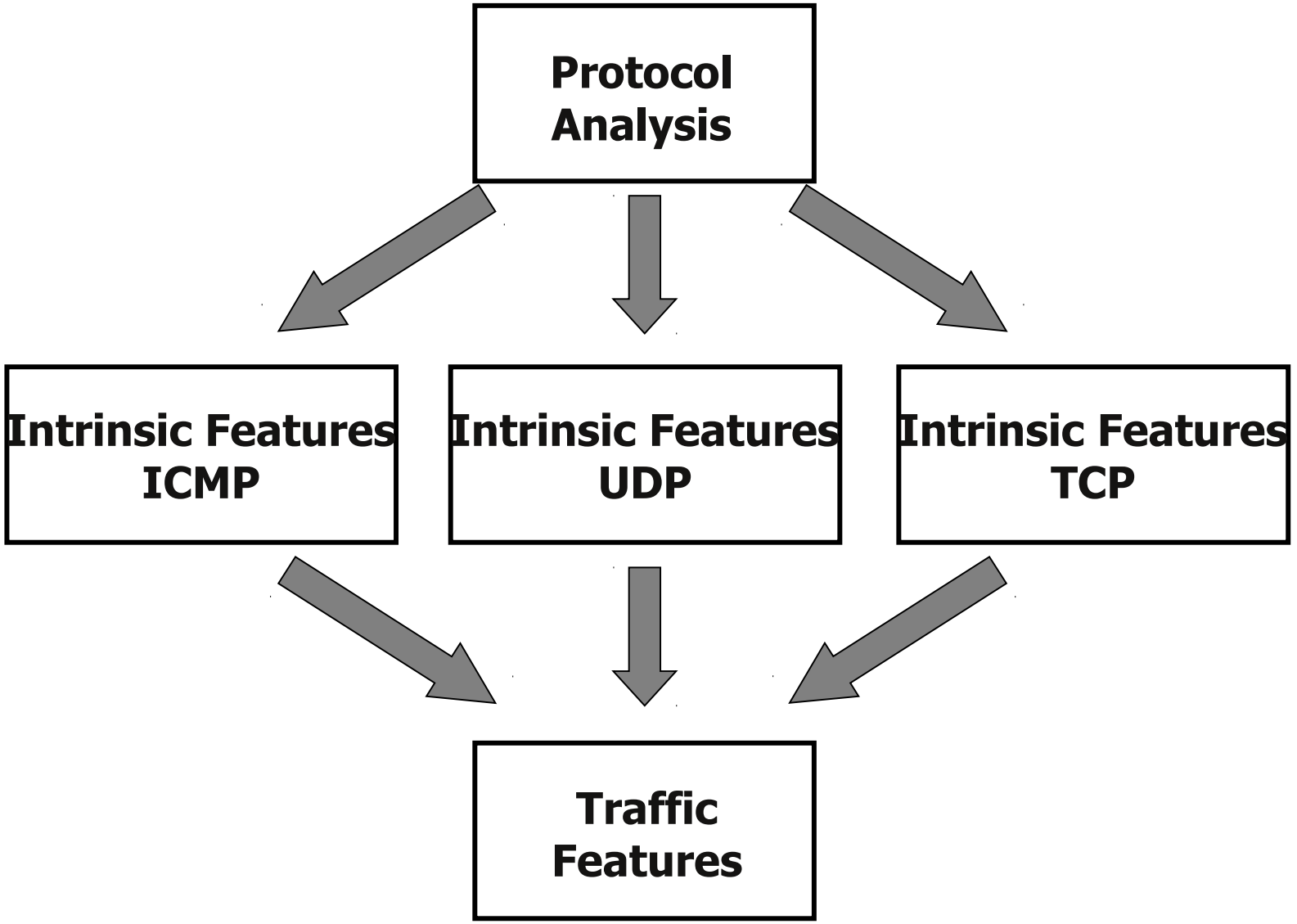
- Network features are based on the “connection” definition
  - UDP and ICMP traffic: single packet
  - TCP traffic: classical meaning
- Intrinsic features
  - duration, protocol type, etc.
- Traffic features
  - Same-service vs Same-host
  - # of connection to same host/service destination as the current one, etc.
  - Time-based vs Host-based
- Content features
  - # of failed login attempts, # of file creations, etc.

* S.J. Stolfo et al. “A framework for constructing features and models for intrusion detection systems”, ACM Transaction on Information and System Security

Features extraction process

- Off-line computation
  - All the packets are already available before the process begins
  - Data can be processed and ordered in the most convenient way
- On-line computation
  - Only past packets are available
  - Data can’t be either organized or ordered
- What we need:
  - Information about the state of the connection which the packet belongs to
  - TCP state diagram emulation
  - Information about packets and connections sharing common properties with the current packet and connection
  - Retrieving the aforementioned information as fast as possible

Features extraction algorithm



Feature Selection

- Each packet is associated to a connection feature vector of 26 elements
- Such a representation may be redundant
- Feature selection
  - Reduce vector space cardinality
  - Reduce the number of computed classification criteria
  - Enhances the speed of detection
  - Selection criteria
  - Sequential forward selection
  - Estimated minimal error probability

Used Features

- 3: service
- 18: dst_host_srv_count
- 20: dst_host_diff_srv_rate
- 1: duration
- 8: count
- 14: same_srv_rate
- 22: dst_host_srv_diff_host_rate
- 19: dst_host_same_srv_rate

Samples Collection

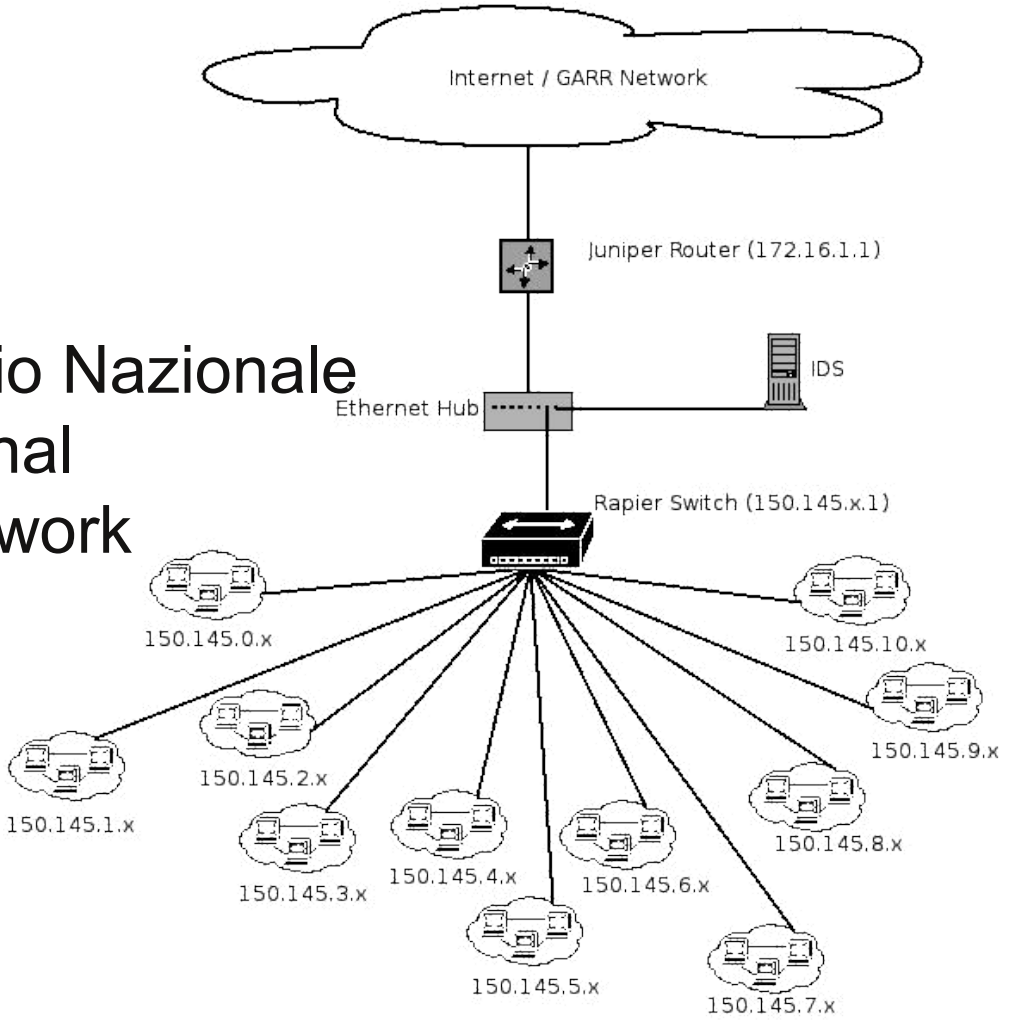
- Raw data is not samples
  - Lack of pre-classification
- Packet labelling
  - After the pre-classification, each training packet is classified as “normal” or “attack”
  - “a-priori” knowledge paradox
  - To detect anomalies, we need to know already what is anomalous
- Pre-classification
  - Snort
  - ISS
  - Human expert
  - Non completely reliable
  - #normal_pkts >> #attack_pkts
- Unreliably classified samples
  - Test 1: labelled as normal
  - Test 2: filtered and deleted from training set

Data Collection 1/2

- Synthetic Network Traffic
  - Network Emulation
    - Unscalable
  - Network Simulation
    - Unrealistic data
- Real Network Traffic
  - Sniffed from the wire
    - Unpredictable
    - (Partially) Uncontrollable
    - Inherently “real”

Data Collection 2/2

- Sniffed Traffic
  - ✓ Genova CNR (Consiglio Nazionale delle Ricerche – National Research Council) network
  - ✓ 16 Mb/s
  - ✓ ~10^6 packets

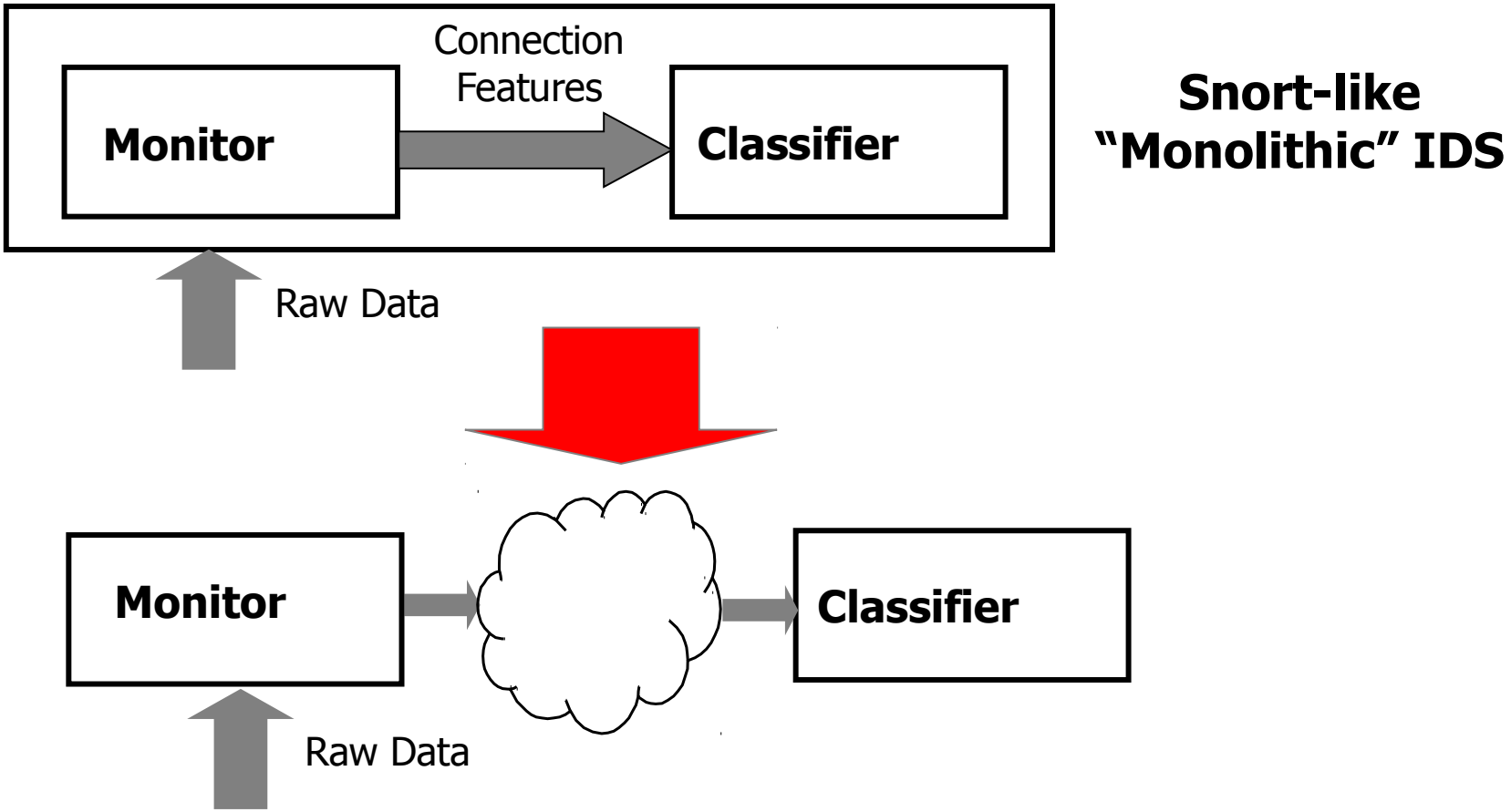


Behavior Modelling

- Anomaly detection
  - Model normal behavior
  - What is not normal is deemed anomalous
- Extended anomaly detection
  - Use both normal and anomalous packets to train the system
  - Obtain classification criteria, to be used in real-time operation
- Used algorithm
  - Boosting
    - Slipper

From “Monolithic” to Distributed IDS

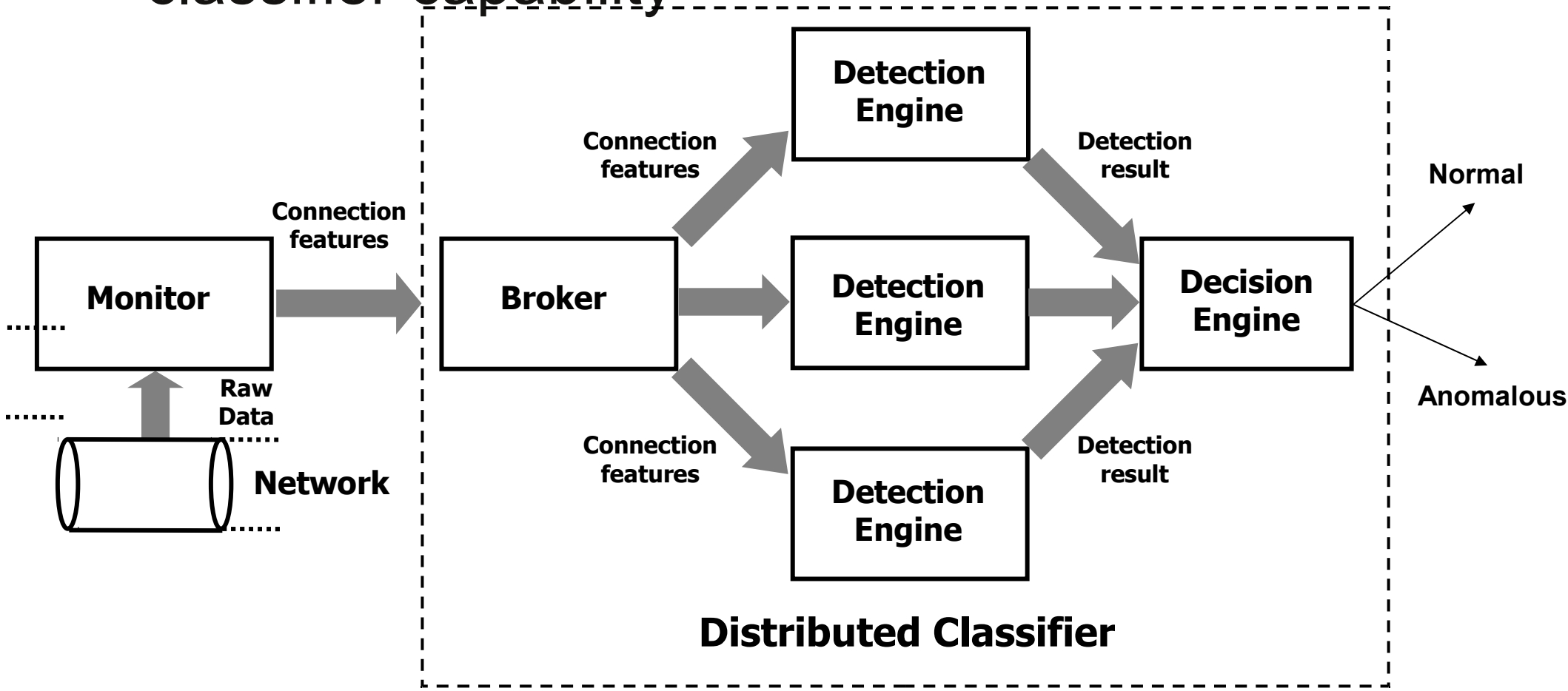
- Reduce packet loss
- Increase scalability



Snort-like “Monolithic” IDS

DIDS: A possible approach

From distributed monitoring to distributed classifier capability



Botnet Detection

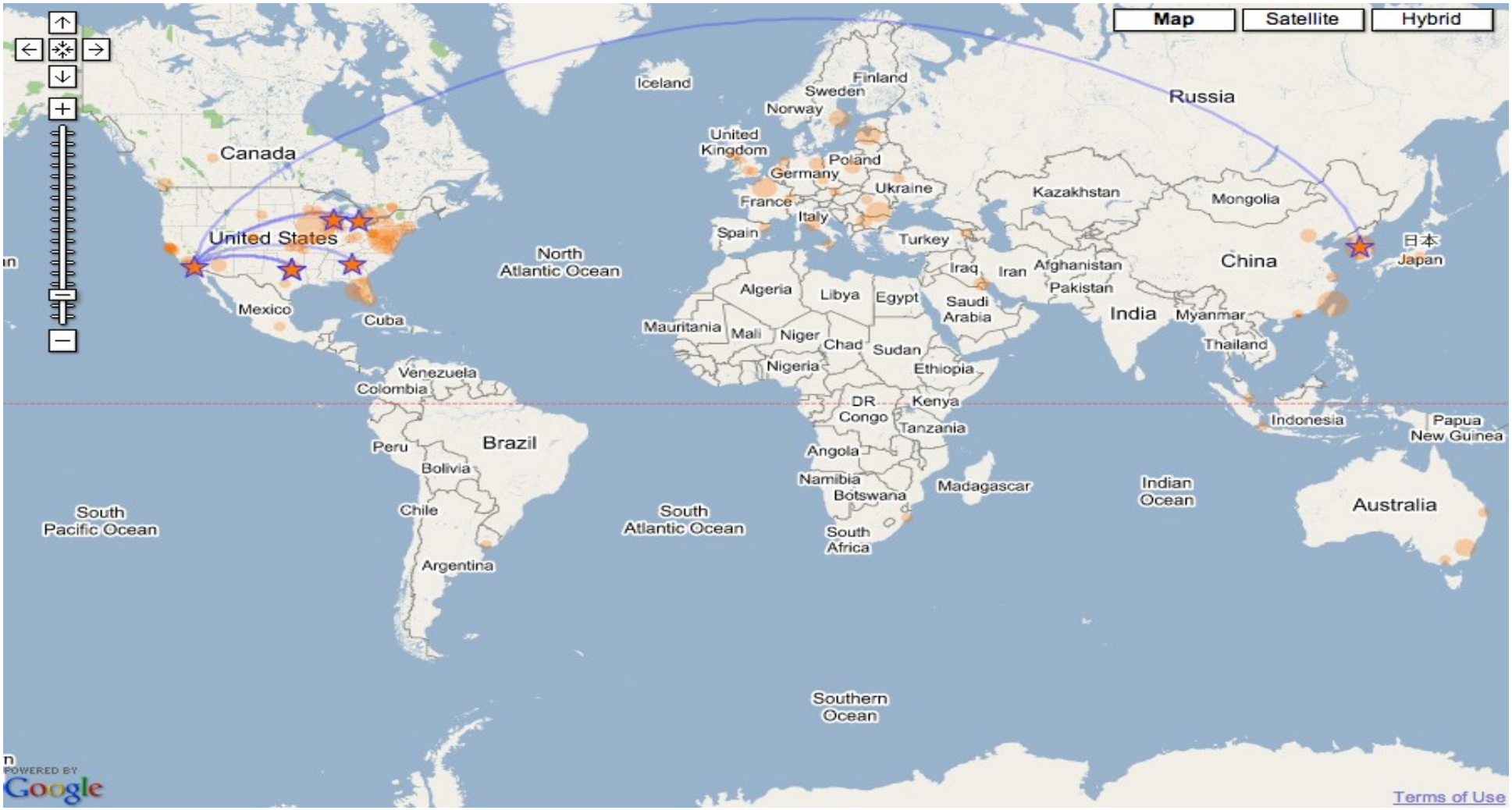
What is a Botnet

- A network of infected hosts, named *bots*, under the control of an operator named *botmaster*
- Control performed by using a *Command & Control* channel
  - Centralized (e.g. IRC, HTTP, ...)
  - Distributed (e.g. P2P...)
- Commands out of a quite large and flexible set can be issued by the botmaster to each bot

Motivation of this work

- Botnets keep spreading
- Botnets are able to perform many malicious actions
  - Spam
  - ID theft
  - Clickfraud (e.g. Google AdSense abuse)
  - Cracking
  - Malware spreading
  - DDoS
  - Traffic Sniffing
  - Keylogging
  - Polls/statistics manipulation
  - …
- Botnets involve economic interests
  - More dangerous than older attack types

The botnet phenomenon

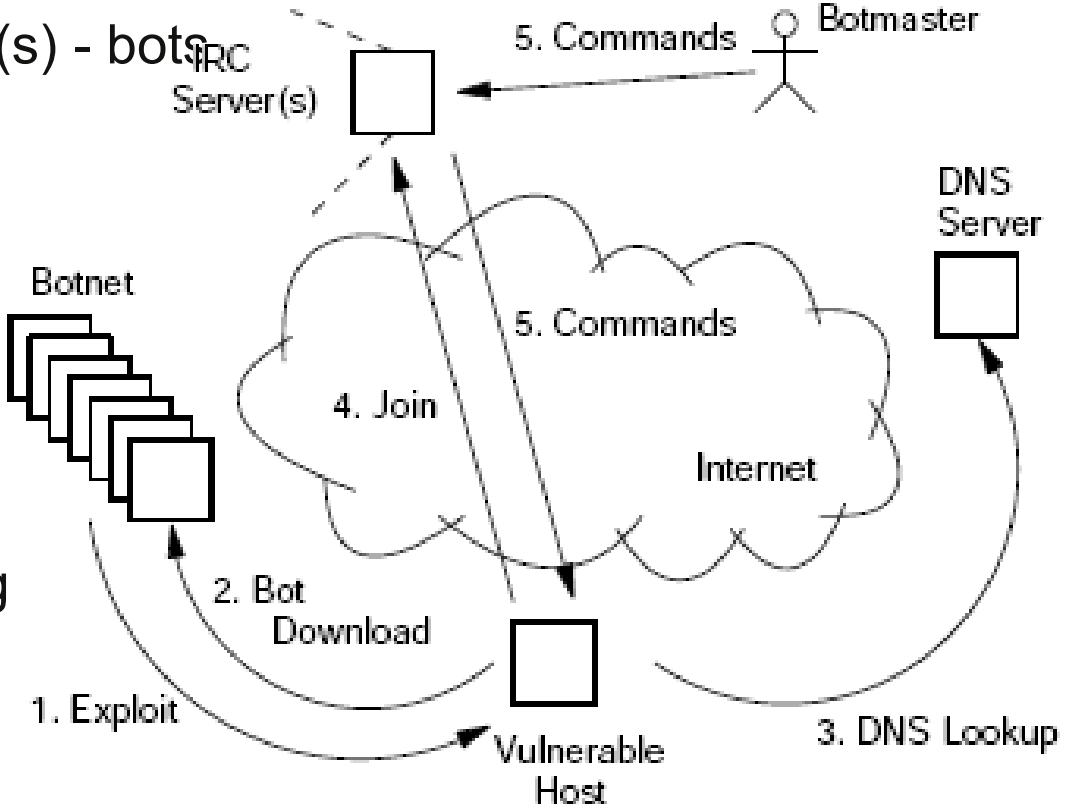


Contribution

- Definition of a model of normal and botnet-related IRC channel usage
- Definition of an architecture exploiting such a model for botnet detection
- IRC user behavior classification aimed at botnet detection by means of pattern recognition techniques

Centralized botnet's lifecycle

- Bot-herder configures initial bot parameters and C&C details
  - register IP at DNS for rendezvous
- bot-herder launches or seeds new bot(s) - bots spreading, botnet growing
  - Vulnerability discovery and exploitation
  - Malicious code download
  - DNS lookup for rendezvous
  - Join the C&C
  - Receive commands from the Botmaster
- losing bots (stasis), botnet not growing
- abandon botnet and sever traces
- unregister DDNS



Botnet Statistics

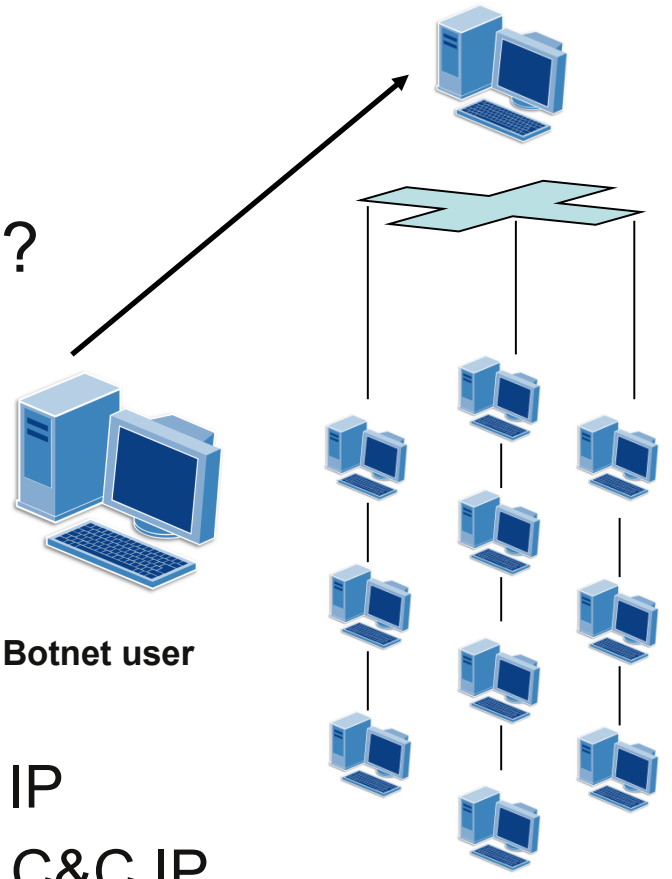
- 60% are IRC bots
  - 70% of all the bots connect to a single IRC server
- 57,000 Active Bots per day for the first 6 months of 2006 (Symantec)
- 4.7 million distinct computers being actively used in Botnets
- Most Botnets are managed by a single server (up to 15,000 bots)
- Mocbot seized control of more than 7,700 machines within 24 hours

Why IRC?

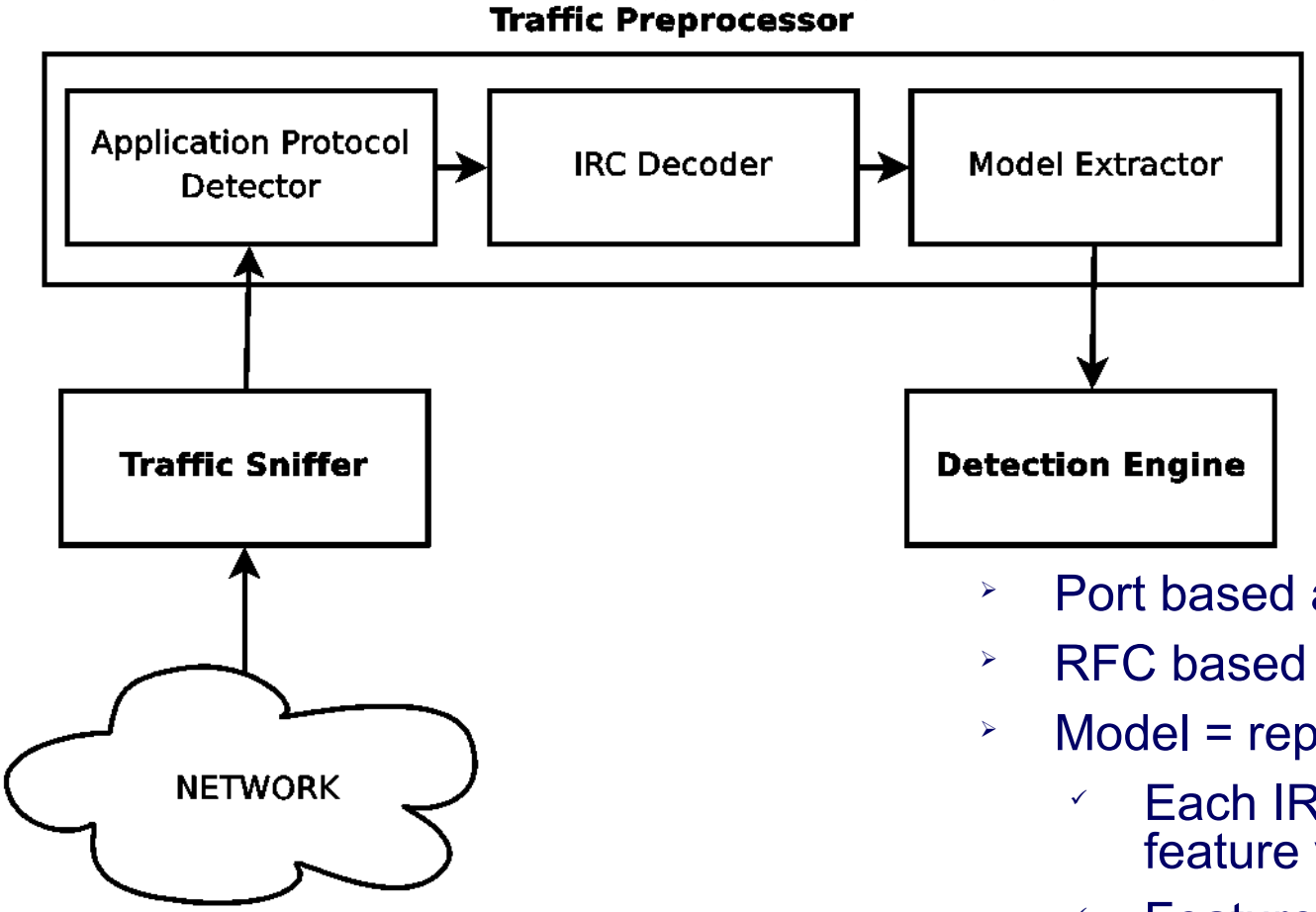
- Oldest and most popular IM
  - Bots were commonly user by channel operator for management and monitoring purposes
- Not owned by anyone – public
  - Defined in RFC 1459
- Text based
- Designed for both point-to-point and point-to-multipoint communication
  - one-to-one, or one-to-group chat
- flexible, open-source protocol
- Potentially able to manage a high number of clients
- Grants anonymity for the botmaster

Centralized C&C

- Easier to manage and use
- Easier to disrupt
- How do the bots know where the C&C is?
  - Hardcoded IP based rendezvous
    - easily uncovered
    - C&C needs replacement after disruption
    - All Bots need replacement
  - Domain names used for rendezvous
    - DNS RR can be updated to current C&C IP
    - Bots can dynamically point to the correct C&C IP



Reference framework



- Port based application protocol detection
- RFC based IRC decoder
- Model = representative features
  - Each IRC channel is represented by a feature vector, representing its status
  - Feature vectors are updated at each event occurring in the corresponding IRC channel

Intuitions about IRC based botnets

- Bursty channel activity
  - After command is issued, bots may respond at once, then be quiet
- Limited vocabulary
- Sentence structure
  - May resemble a shell command
  - The same recurring structure may be found in many sentences
- Disproportion between user and control activity in a channel
- “strange” words used for communication
  - Disproportion of consonants and vowels in words used for chatting
    - Language dependent
- Changes and structure of chat room topic
- Unusual nicknames
  - Completely random OR
  - Unexpextedly regular

IRC channel attributes

- Users Number:
  - total number of users in the channel
- Average words number:
  - average number of unique words in a sentence
- Averange/Variance of Channel Dictionary Cardinality:
  - Mean and variance of the vocabulary's cardinality
- Unusual Nicknames*
- Equal Answers:
  - number of sentences with a common ordered subset of words
- Control Commands Number:
  - count of channel control commands issued
- Join Number:
  - JOIN rate in the channel
- SetMode Number:
  - SetMode rate in the channel
- Nickname Changes:
  - count of nickname changes in a channel
- Ping Number:
  - PING rate in the channel
- IRC Commands Number:
  - overall IRC command rate
- Active Users Number:
  - number of users active in the channel

*J. Goebel and T. Holz. Rishi: identify bot contaminated hosts by irc nickname evaluation. In *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pages 8–8, Berkeley, CA, USA,* 2007. USENIX Association.

Experimental Setup

- Data collection
  - Botnet related traffic from the Georgia Institute of Technology network
  - Normal IRC chats logged from the University of Napoli network
- Three datasets
  - 50,000 samples (25,000 normal + 25,000 botnet-related)
    - Small, evenly split
  - 149,999 samples (75,010 normal + 74,989 botnet-related)
    - Large, evenly split
  - 165,000 samples (150,000 normal + 15,000 botnet-related)
    - Large, more realistic distribution of t-uples
- Selected algorithms
  - SVM (Support Vector Machine) – very “popular”
  - J48 (Decision Tree) – very “quick”
- Performance evaluation
  - 10-fold cross validation

Traffic Classification

Traffic Identification

- Need to associate flows to the applications that generate them
- {UDP, IP_{SRC} :10.0.0.1, $PORT_{SRC}$:31215, IP_{DST} :212.48.72.19, $PORT_{DST}$:80} => SKYPE!
- {TCP, IP_{SRC} :10.0.0.1, $PORT_{SRC}$:2233, IP_{DST} :13.29.10.199, $PORT_{DST}$:25} => SMTP!

  - Mellia et al., “Traffic classification and its applications to modern networks”, Elsevier Computer Networks, Dec. 2008
  - Callado et al., “A survey on internet traffic identification”, IEEE Communications Surveys & Tutorials, July 2009.

Motivation

- What if we cannot classify traffic?
  - We have no clue of what our links carry
    - How is people using the Internet?
    - What's the killer application?
    - Does it really matter to model this or that?
    - Is something "strange" happening and we don't know it?
  - We cannot
    - do provisioning
    - perform resource allocation and offer QoS
    - enforce security policies (e.g. Firewalling)
    - do accounting based on typology of traffic
    - study network traffic if we cannot retrace phenomena to specific applications and protocols (e.g. congestion)

Approaches

- Port-based
  - - Very inaccurate
  - + Simple & Fast
  - + Privacy-friendly
- Deep Packet Inspection (DPI)
  - + Accurate
  - CPU intensive
  - Doesn't care about Privacy
  - Losing accuracy with increasing use of encryption and protocol obfuscation
- Pattern Recognition
  - + Good Accuracy
  - +- Slow/Fast
  - Privacy Friendly
  - More robust to encryption and obfuscation

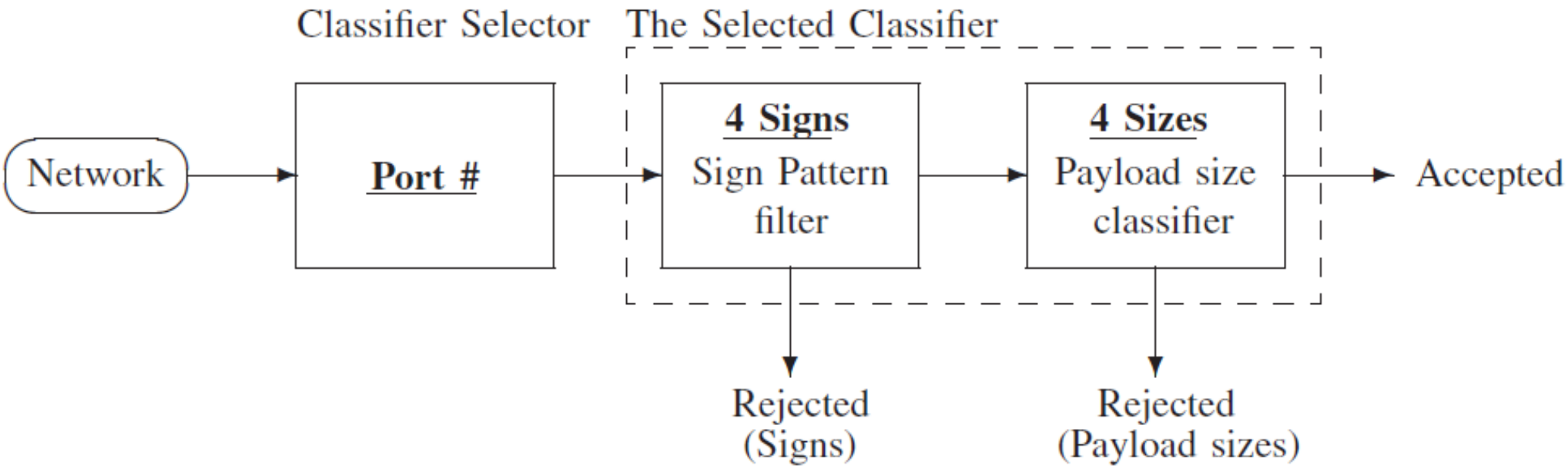
*Nguyen et al. "A survey of techniques for internet traffic classification using machine learning". IEEE Communications Surveys and Tutorials, 2008)

The technique here evaluated

- Designed to be fast
  - Few and early available features
  - Ensemble of Classifiers
  - Decision Trees algorithms
- Thought to discover flows using well-known ports assigned to other applications
- Applicable e.g. to security policy enforcement, but also accounting, QoS, etc.

Classifier Selection Ensemble

- The Ensemble is made of Experts and an Oracle
- The Oracle uses ports to determine the regions of competence
- Each Expert is a two stage classifier



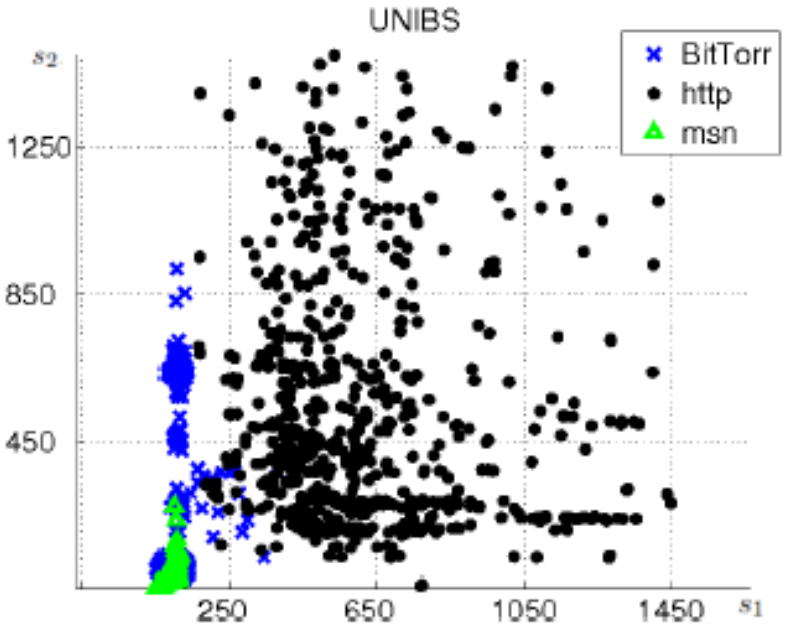
1st Stage: pkt directions

- We call packet directions “signs”:
  - 0: server to client
  - 1: client to server
- The pattern of the 1st four packets (carrying payload) is used to
  - Reject packets
  - Assign a classifier from the 2nd stage

Signs	Protocol and port number					
	POP3	FTP	SMTP	msn	BitTorr	HTTP
123 4	110	21	25	1863	6881	80
000 0	0	138	16	0	0	3
000 1	1	75	55	0	0	0
001 0	21	216	543	0	0	0
001 1	0	0	4	0	1	0
010 0	749	21	604	1	0	0
010 1	18823	5845	18186	0	1	0
011 0	17	1	18	0	1	0
011 1	0	0	1	0	0	0
100 0	0	0	0	328	23	5348
100 1	0	0	0	30	520	240
101 0	0	0	0	660	3609	826
101 1	0	0	0	4	753	12
110 0	0	0	0	1	8	427
110 1	0	0	0	0	87	76
111 0	0	0	0	0	9	108
111 1	0	0	0	0	45	23

2nd Stage: payload sizes

- Each sign combination identifies a different classifier
  - Separate training (easily retrainable)
  - Better accuracy
- Features: 1st four payload sizes
- Decision Tree algorithms
  - Fast
  - No assumptions on probabilistic distribution of data



Data sets

- Training traces:
  - CAIDA 2002
  - LBNL 2004
  - UNIBS 2007

TABLE II

NUMBER OF FLOWS IN THE THREE TRAINING DATA SETS.

Protocol	Port	UNIBS	CAIDA	LBNL
POP3	110	19611	9591	1172
SMTP	25	19427	11831	20825
HTTP	80	7063	5930	81984
FTP	21	6296	1652	–
BitTorrent	6881	5057	–	–
msn	1863	1024	–	–
netbios-ssn	139	–	4575	–
HTTPS	443	–	25427	18013
oms	4662	–	–	1716
IMAP4	993	–	–	7677

- Testing traces:
  - UNINA 2004
  - UNINA 2009

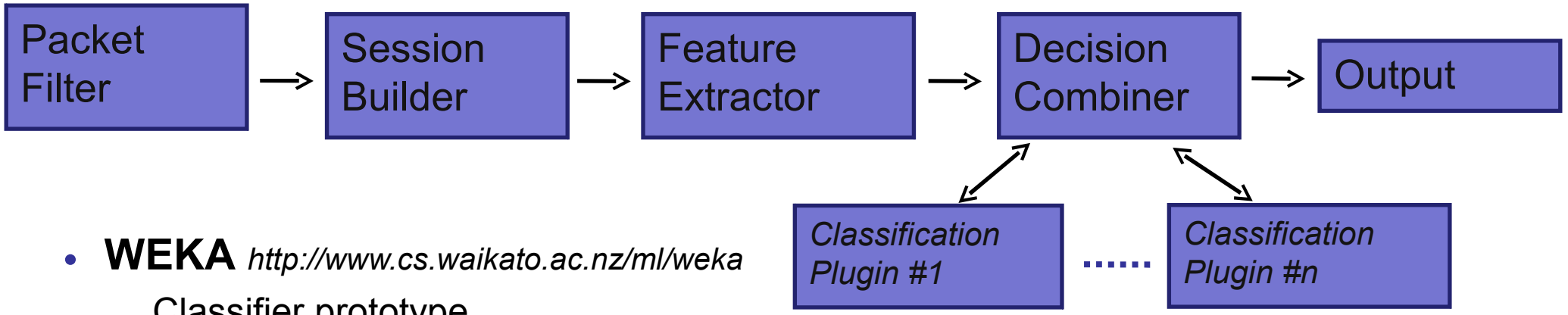
TABLE III

NUMBER OF FLOWS IN THE UNINA DATA SETS USED FOR TESTING.

Protocol	Port	UNINA2004	UNINA2009
HTTP	80	506795	144042
non-HTTP	80	2245	803

Tools

- **TIE** – Traffic Identification Engine *http://tie.comics.unina.it*
  - Process and filter traces
  - Aggregate packets into bidirectional flows
  - Extract features
  - Ground-Truth with *L7-Filter* DPI classification plugin
    http://l7-filter.sourceforge.net



- **WEKA** *http://www.cs.waikato.ac.nz/ml/weka*
  - Classifier prototype

Analysis of rejected traffic

- We analyzed TCP 80 traffic labeled by our identification system as '*rejected*'.
  - All the correctly accepted biflows were actually related to HTTP traffic (*94% GET, 4% POST,* …)
  - Several rejected biflows were generated by peer-to-peer applications (*eDonkey, Bittorrent, WinMX,* …)
  - Up to 50% of the non-HTTP biflows could not be ascribed to an application. We manually verified that they did not exchange HTTP traffic. Undisclosed proprietary protocols!
  - Correctly rejected traffic accounted for >5% of the packets (*with no port filtering enforced in the observed network!*)

VoIP Security
Social Threat Detection

Social Attacks in VoIP networks

- SPIT – SPAM over Internet Telephony
  - Mainly for commercial purpose
- Vishing – Voice Phishing
  - Stealing critic information leveraging customers trust on Telephony Service
- Why VoIP networks?
  - Lower cost with respect to PSTN
  - Parallelism (botnet)

Some SPIT attacks

- INVITE flooding
  - Columbia University, 2007
- Registered advertisements
  - Skype calls interrupted by another call playing a registered message, 2008
- Vishing
  - Calls redirecting to a fake bank website to steal bank codes, USA, 2006

What's the point?

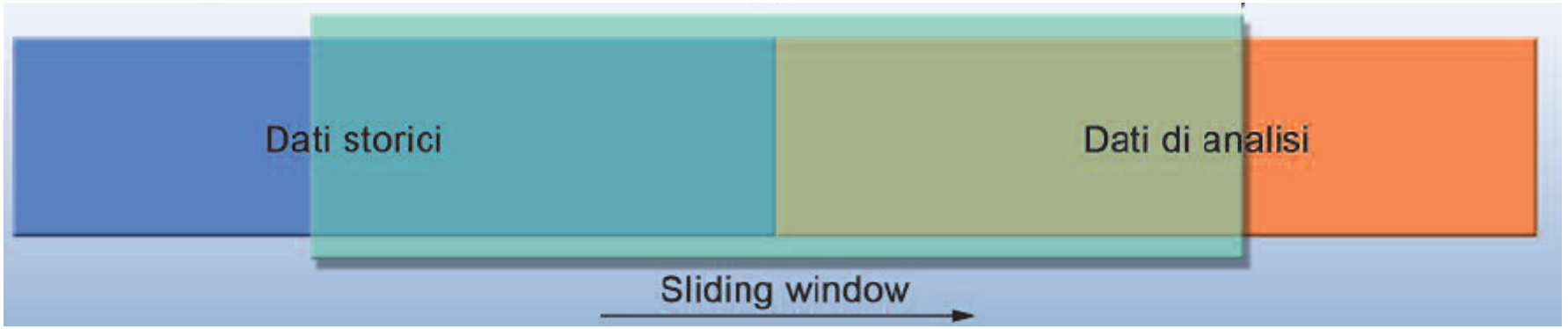
- SPIT callers can behave quite differently from legal callers
  - A legitimate caller typically makes and receives calls, while a spammer makes a large number of calls but seldom receives a call
  - A legitimate caller calls the same number more than once, while a spammer calls as many callees as possible and thus seldom repeats dialing the same number
  - A legitimate caller usually calls their buddies, while a spammer often calls a large number of unknown callees
- Each user is characterized by two personal profiles, both as a caller and as a callee

  …let's use clustering to synthesize different user behaviours

Clustering Detection approach

- Separate “good” callers/calls from “bad” callers/calls in real time
- System prototype

- available information:
  - Call details collected in two months
- Sliding window analysis



Selected Features

- Number of calls in the test period (per user)
  - As a caller
  - As a callee
  - Calls as a caller/calls as a callee
  - …
- Contacts (per user)
  - “just-once” users
  - “just-once” users/tot calls
  - …
- Call duration (as a caller, as a calle)
- Signalling byte amount (as a caller, as a calle)
- Voice traffic byte amount (as a caller, as a calle)
- …

Thanks for your attention!

Any questions?