

Intrusion detection alerts

January 13th, 2004

Hervé Debar

Agenda



- ▶ **Project objectives**
- ▶ **Alert correlation**
- ▶ **IDS sensor architecture**

The supervision project



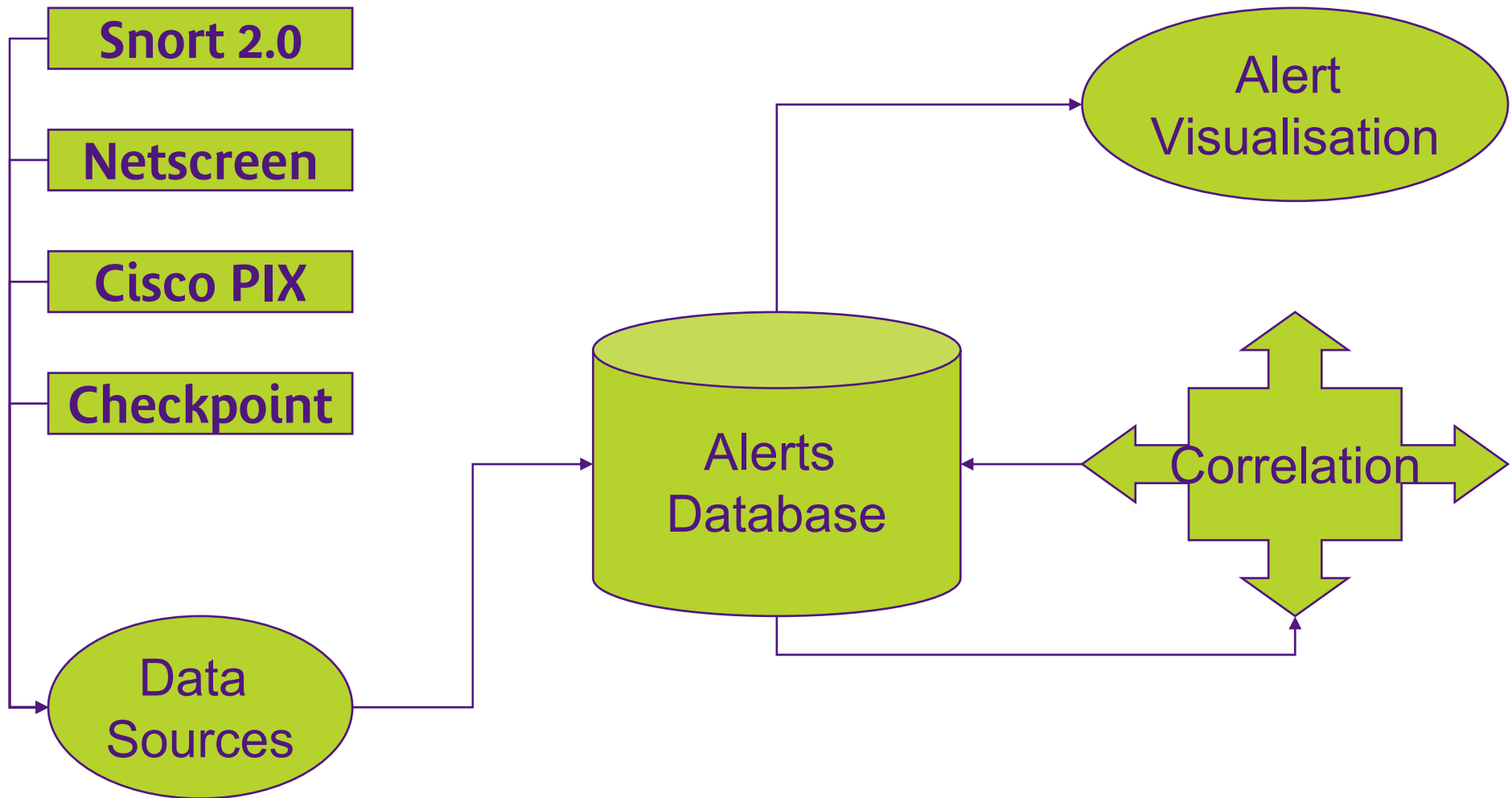
- ▶ **Deploy sensors in sensitive areas of the network**
- ▶ **Centralize log data**
- ▶ **Correlate information from multiple sources**
- ▶ **Present information to security operators**
- ▶ **Propagate alerts to management systems**

Chronology



- ▶ **December 2002: Initial deployment**
 - ▶ Packaging of Snort and ACID
- ▶ **May 2003: ACID patches**
 - ▶ Locked in ACID code
- ▶ **October 2003: version 1.0**
 - ▶ Looses database portability
 - ▶ Better data security
 - ▶ Simpler navigation
- ▶ **December 2003: version 1.0.2**
 - ▶ IDMEF compatible (includes snort package)
 - ▶ Complete packaging and installation procedure

General Architecture



Architecture principles



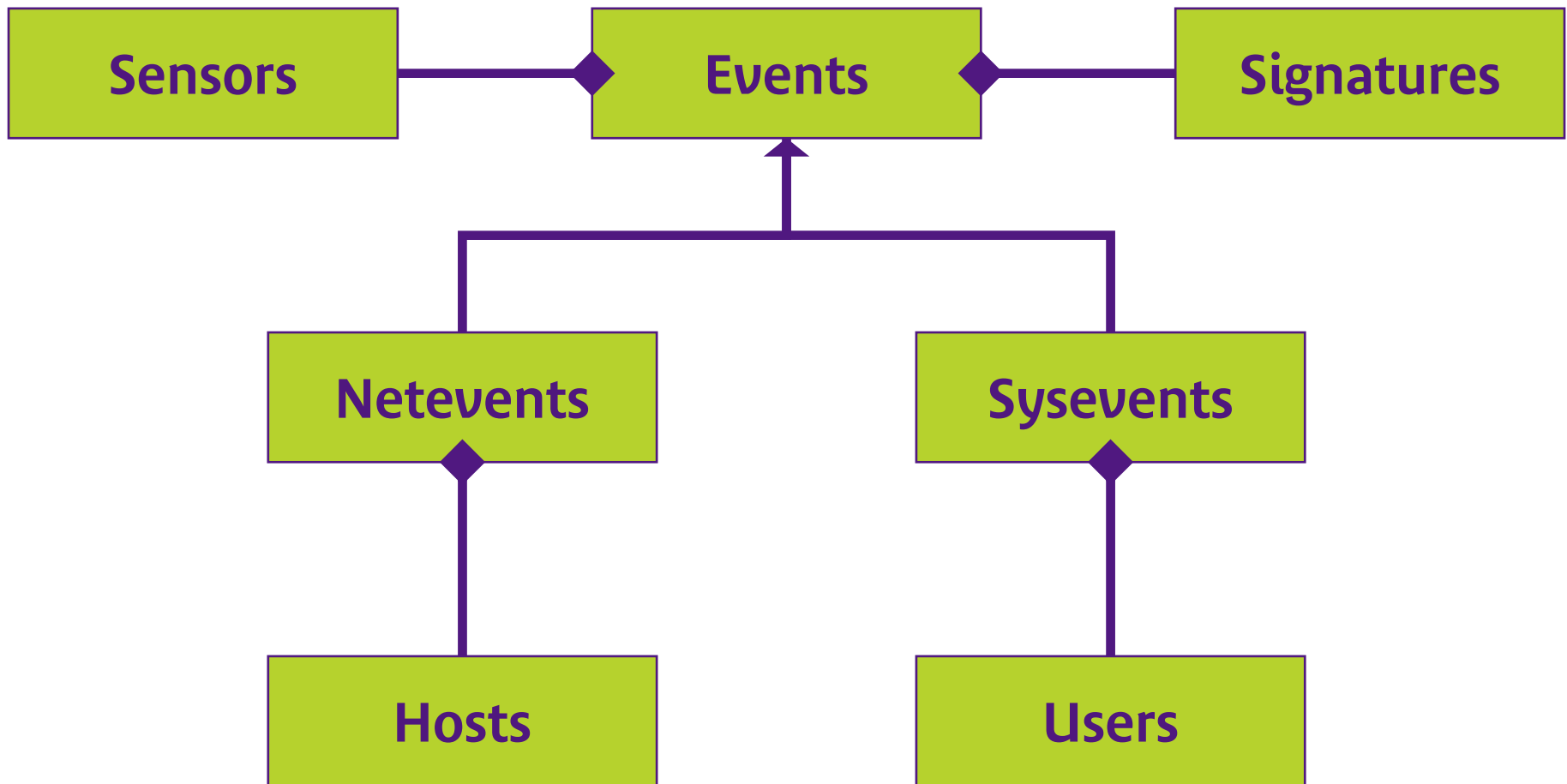
- ▶ **Database centric**
 - ▶ Database acts as persistent, safe data bus
 - ▶ Data sources insert events in database
 - ▶ Correlation processes grab information from DB, push it back
 - ▶ Presentation only takes information from DB
- ▶ **Easy web access**
 - ▶ Dynamic web site
 - ▶ Reports
- ▶ **Multiple independant correlation processes**
 - ▶ {Alerts} to alert
 - ▶ {Alerts,Inventory} to alert
- ▶ **Simple, « universal » database schema**
 - ▶ Very heterogenous log formats

Consequences



- ▶ **Performance linked to database**
 - ▶ Insertion
 - ▶ Complex queries
- ▶ **Naturally distributed application**
 - ▶ Multiple distributed alert databases
- ▶ **Easy to secure**
 - ▶ Transport security: existing protocols
 - ▶ Data security: database + application
- ▶ **Heterogenous log translation**
 - ▶ Events: IDMEF, firewalls, VPN, system
 - ▶ Inventory: Nessus, ...

Database schema



Database schema



▶ Origin:

- ▶ snort 106 (too many tables, network specific)
- ▶ IDMEF (too many tables, imprecise signature)
- ▶ M2D2

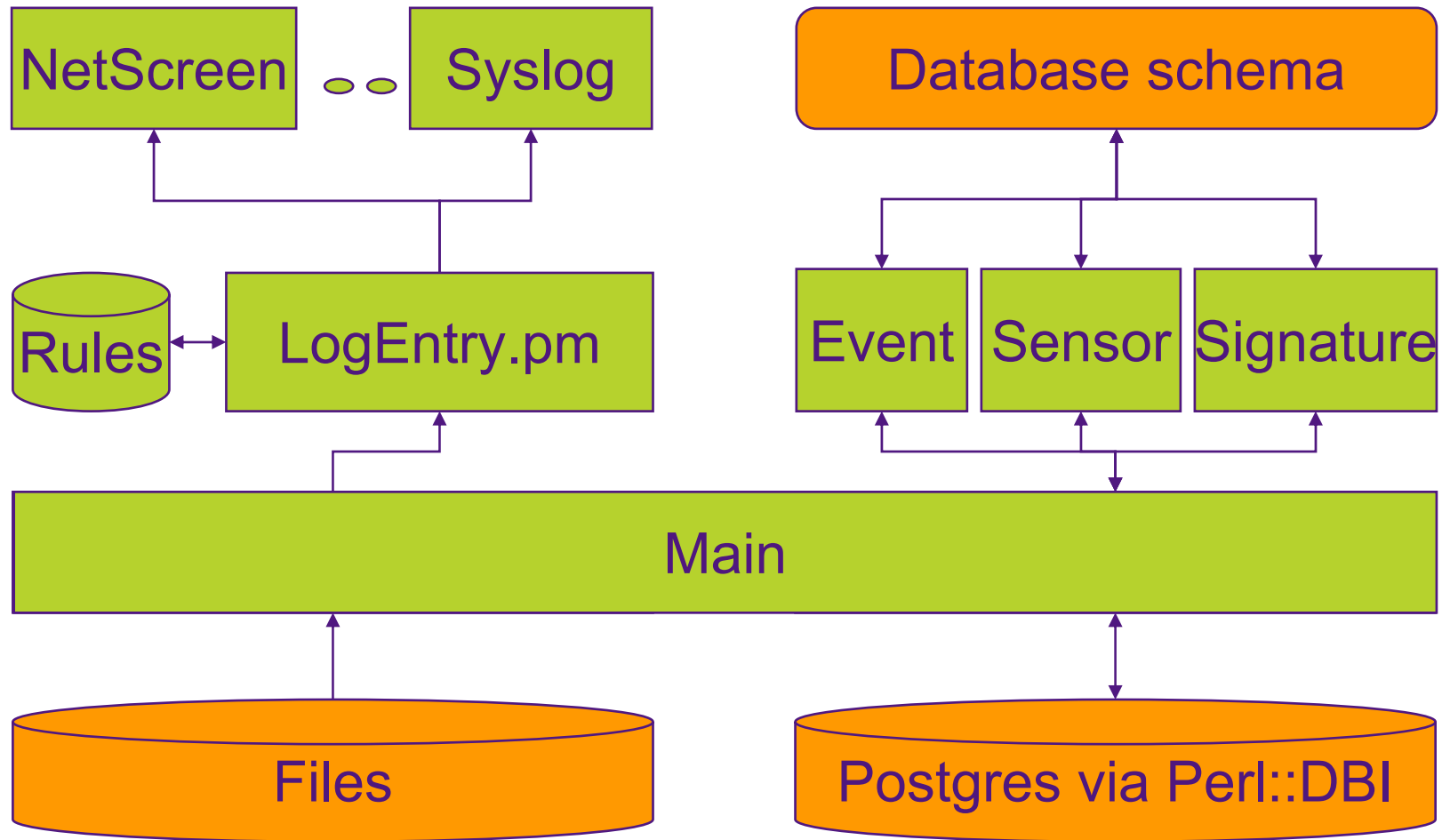
▶ Core: event

- ▶ What: signature
- ▶ Who: sensor
- ▶ When: timestamp

▶ Complement information

- ▶ Network-related information
- ▶ System-related information
- ▶ AdditionalData

Log import





Observations on alerts

- ▶ **False positives**
 - ▶ None really identified
- ▶ **Mischaracterization**
 - ▶ HTTP traffic
- ▶ **No immediate impact**
 - ▶ ICMP
 - ▶ SNMP
 - ▶ Pre-processor messages
- ▶ **Security issues requiring attention**
 - ▶ Daemon infection
 - ▶ Worms
- ▶ **High volume**

Alert Correlation



Purpose of alerts

- ▶ **Countermeasure**
 - ▶ Alert without explanation is useless
 - ▶ Tailored to operator knowledge
- ▶ **Carefully weighted priority**
 - ▶ Depends on signature
 - ▶ Depends on sensor location
- ▶ **Detailed contextual information**
 - ▶ Relevance
 - ▶ Success
- ▶ **Volume limitation**
 - ▶ Tailored to operator capacity

IDS objectives



- ▶ **Number of alerts to handle by operator tends to zero**
- ▶ **Catch malicious events that are not characterized by an identified vulnerability**
- ▶ **Automate reaction to known problematic events**
 - ▶ Identified as malicious
 - ▶ Identified as part of some « correlation process »

Correlation objectives



▶ **Volume reduction**

- ▶ Elimination
- ▶ Fusion
- ▶ Aggregation
- ▶ Synthesis (logic links between alerts)

▶ **Diagnostic improvement**

- ▶ Type of activity
- ▶ Relevance
- ▶ Success

▶ **Activity tracking**

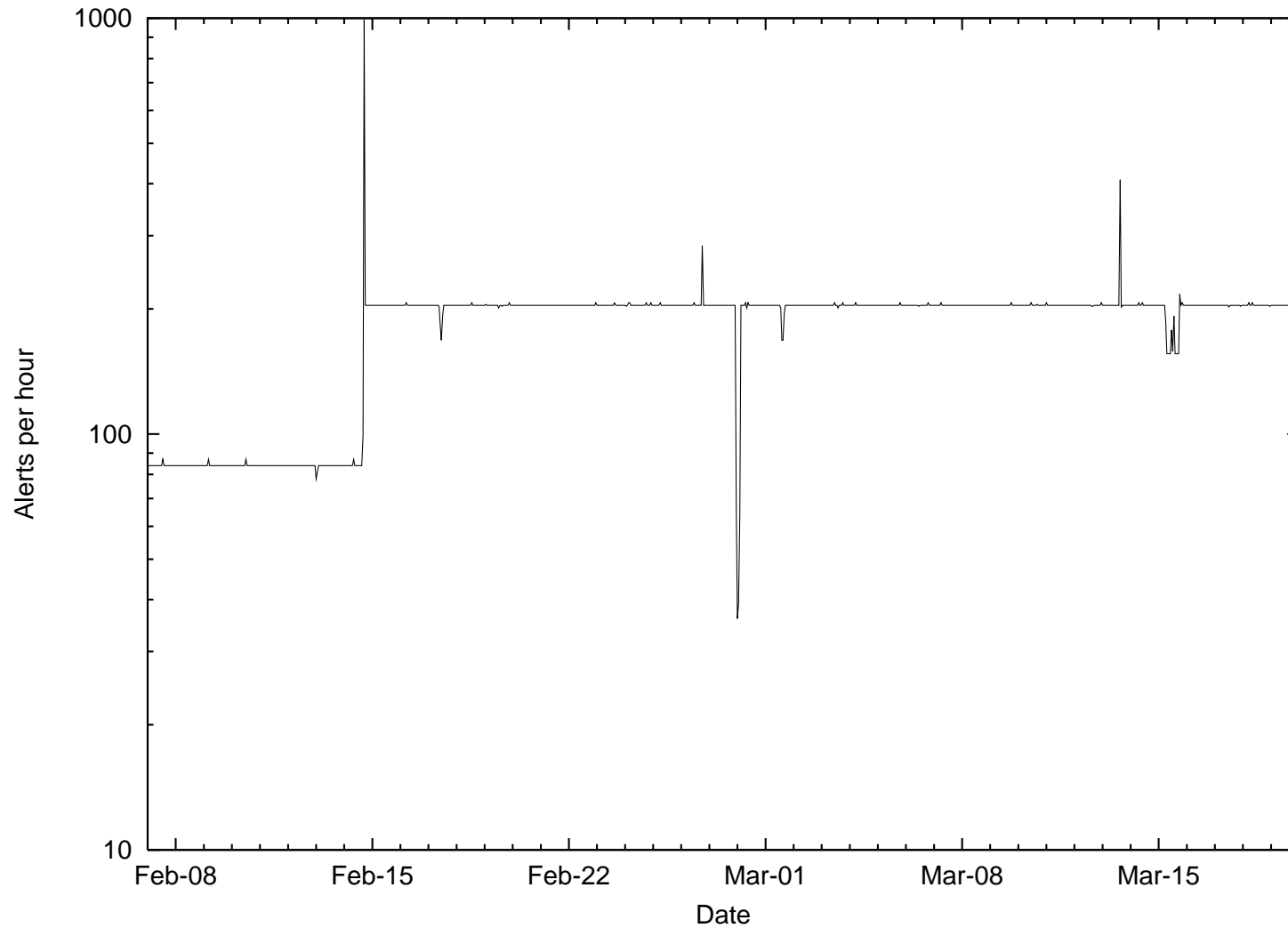
- ▶ Information leaked to attackers
- ▶ Information gathered from attackers

Current implementation

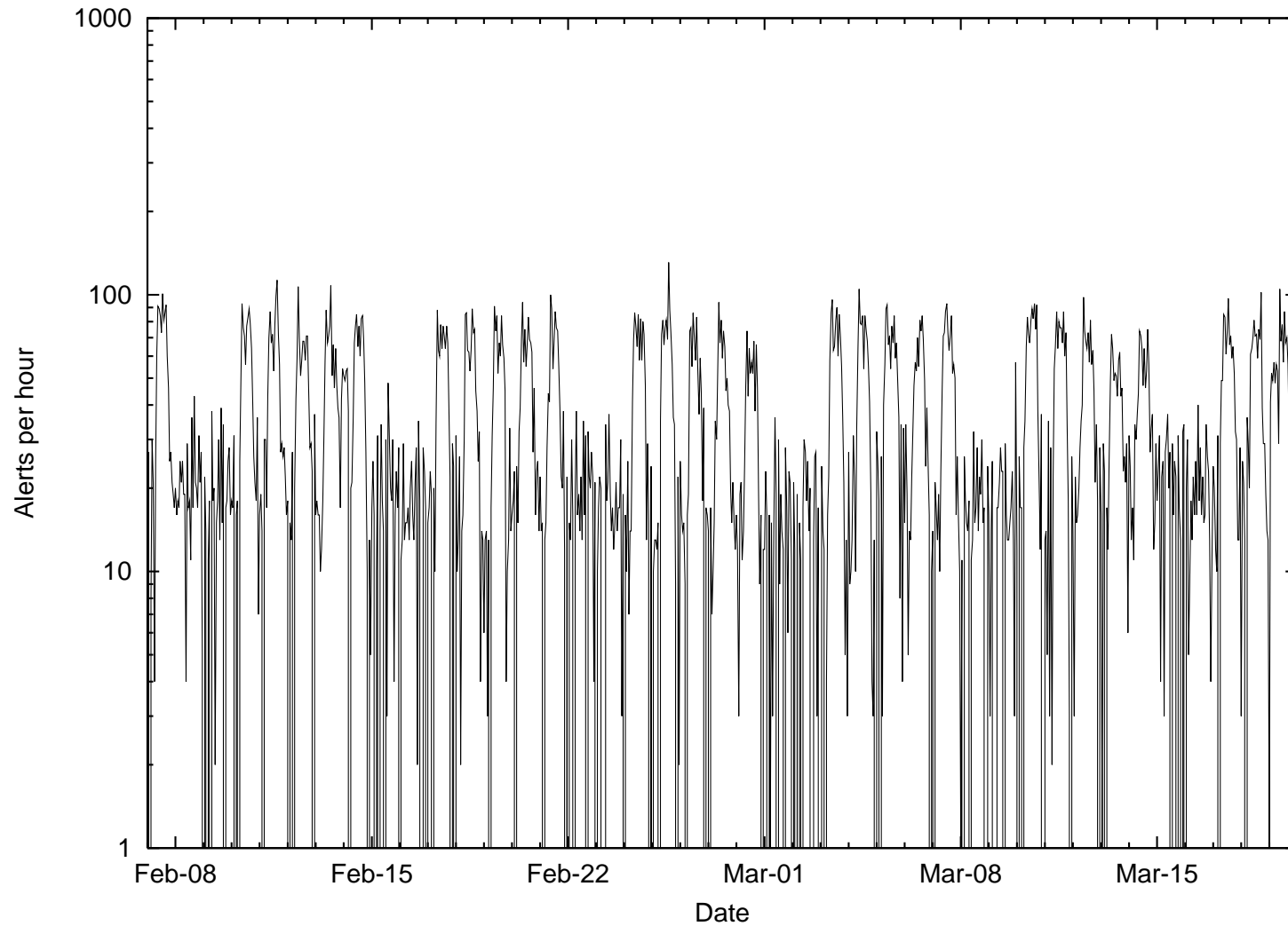


- ▶ **Inventory tracking and assessment**
 - ▶ Classic Nessus vulnerability assessment
 - ▶ Inventory information linked with deductive process
- ▶ **Statistical identification of trends**

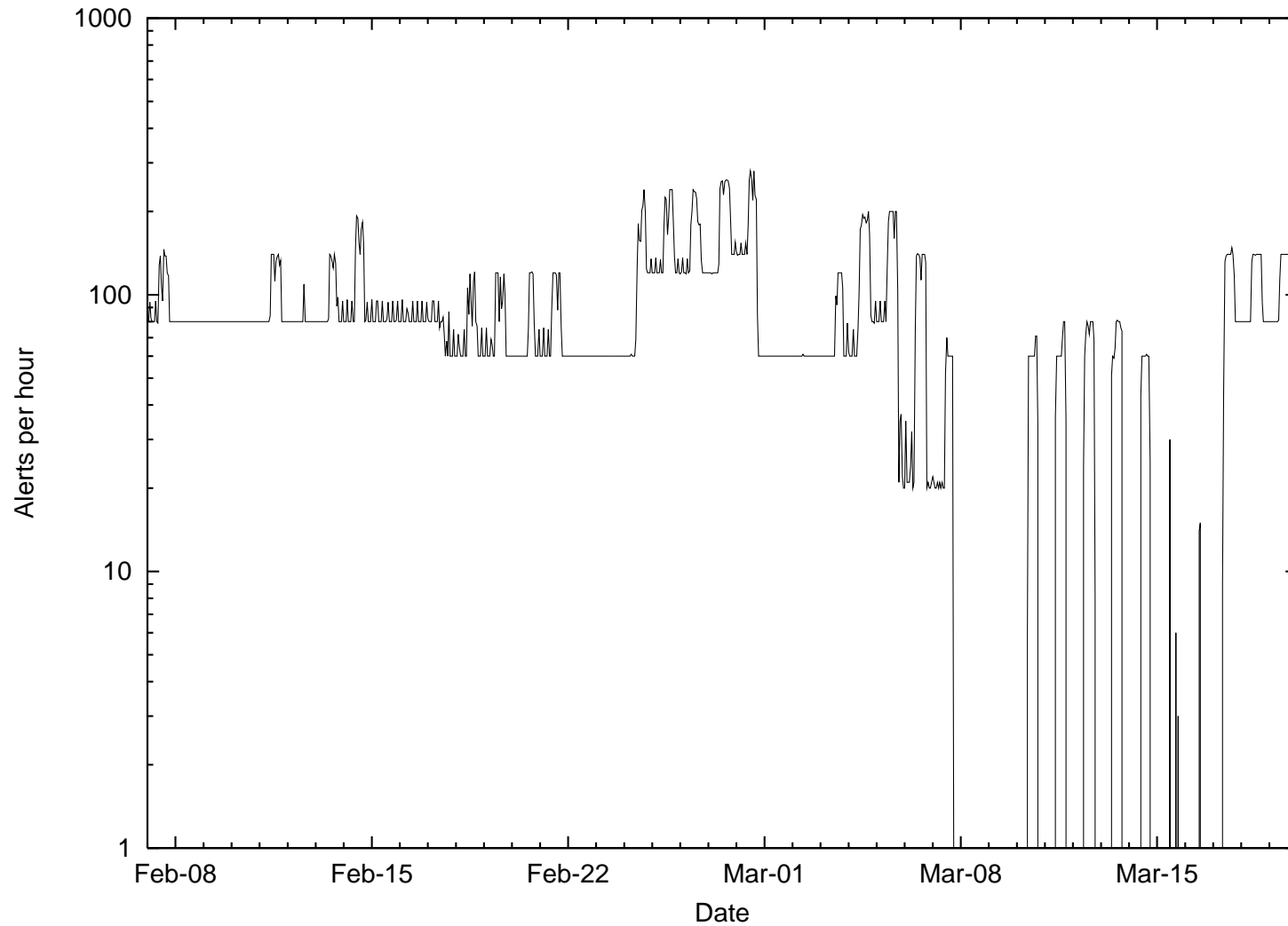
Case 1: clockwork



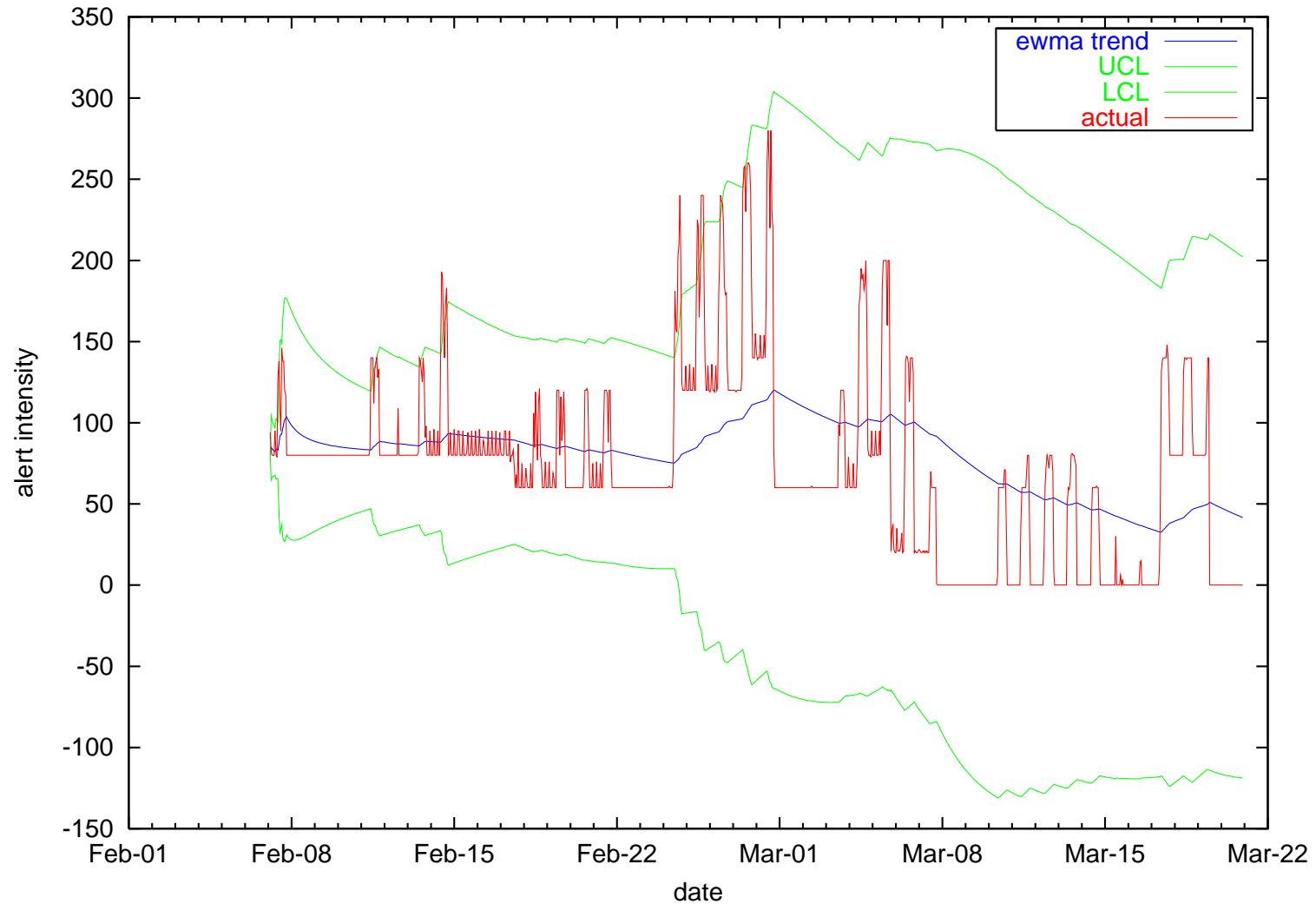
Case 3: fuzzy



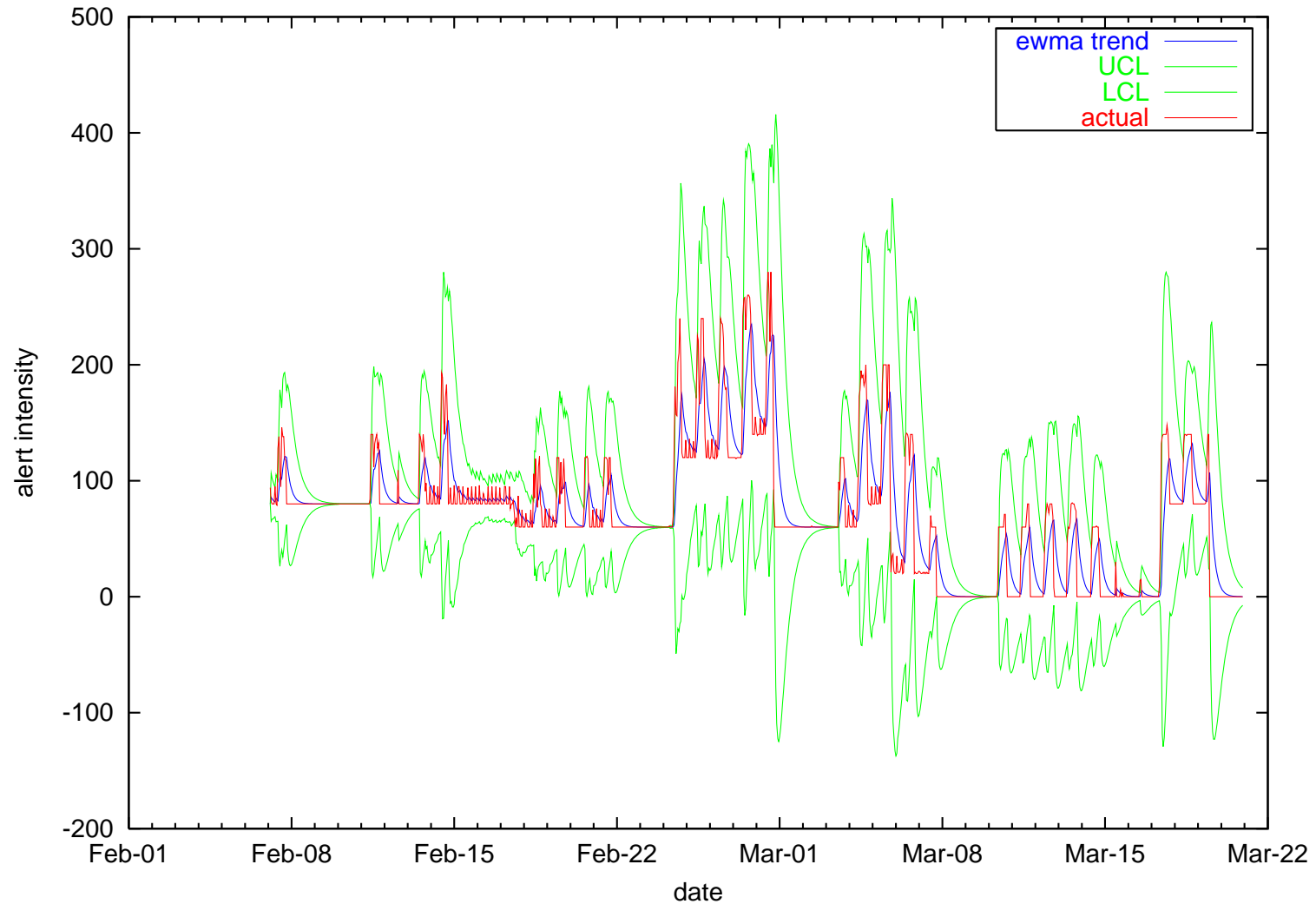
Case 2: plateau



Case 2: enveloppe (long)



Case 2: enveloppe (short)



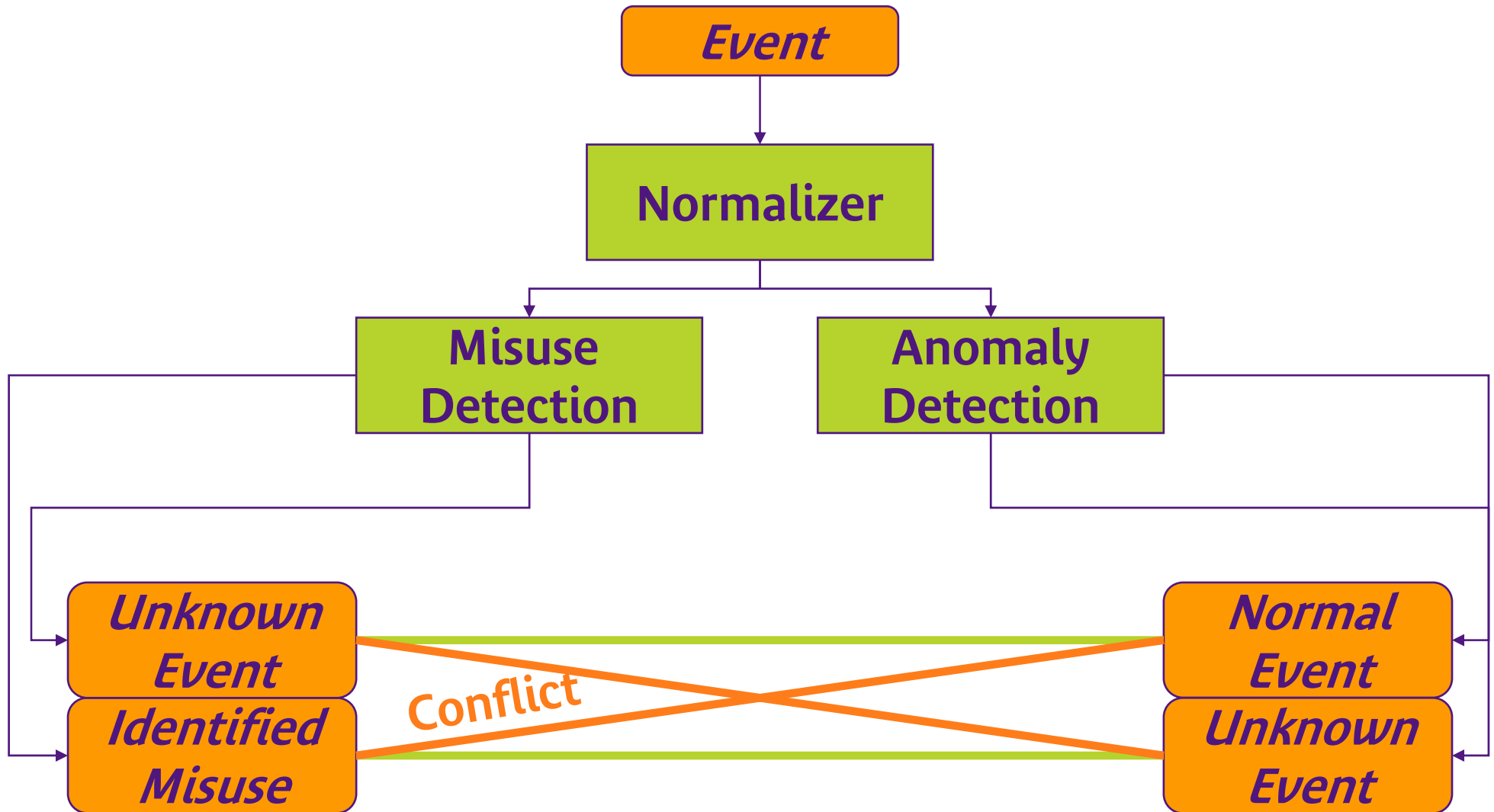
Conclusion on correlation



- ▶ **Must handle high alert volume**
 - ▶ Possible warning of security issues
 - ▶ Response time of a few hours
- ▶ **Unclear what to do with low-frequency alerts**
- ▶ **Still need better information from IDS sensors**
 - ▶ Allow real-time blocks (IPS)
 - ▶ Don't waste time rebuilding information with guesswork
 - IP sessions
- ▶ **Need for better sensors**

Thoughts on sensors

Classic IDS sensor architecture

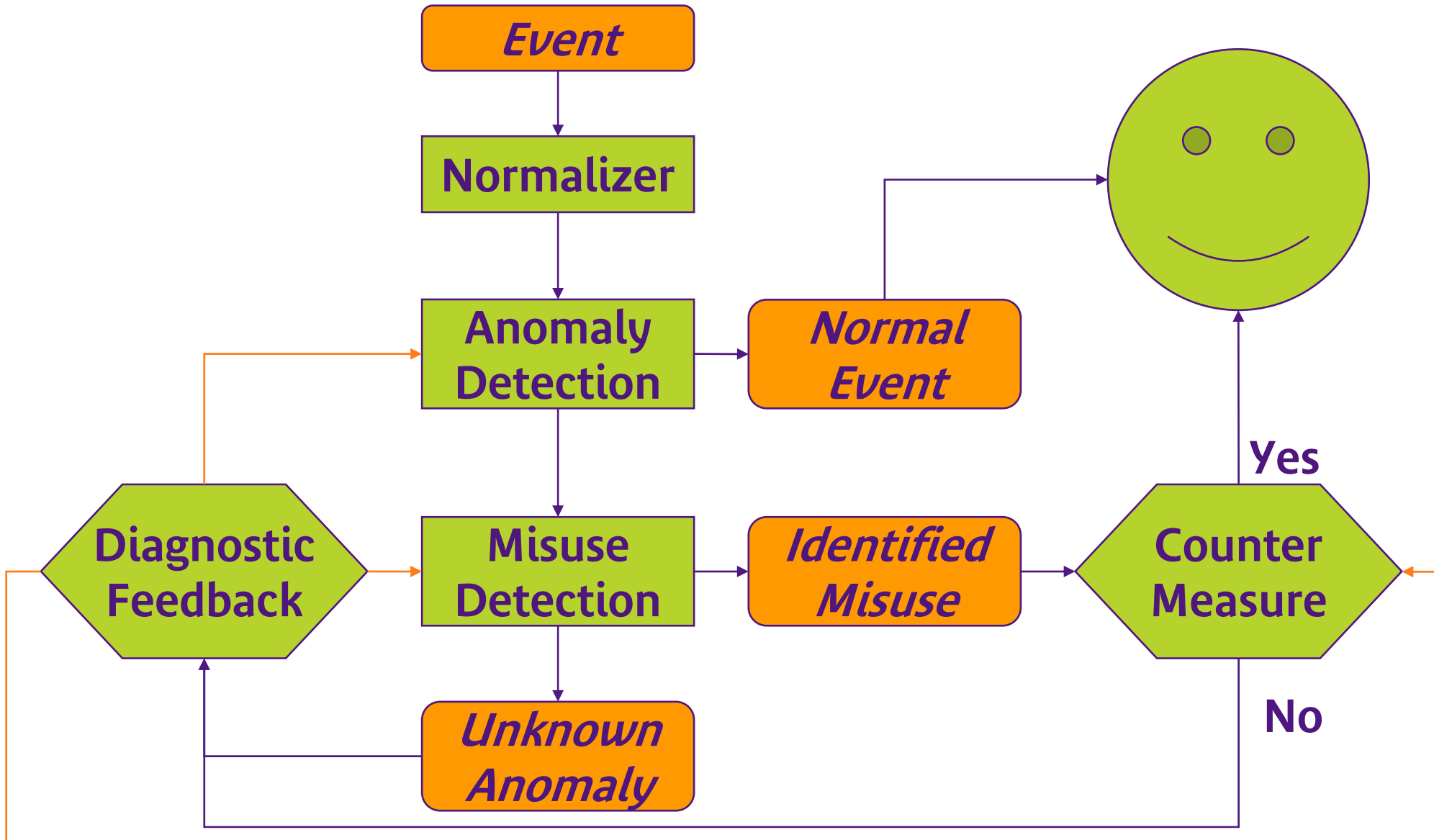


Classic IDS architecture issues



- ▶ **Duplicate analysis has performance cost**
 - ▶ Misuse detection: identify known malicious events
 - ▶ Anomaly detection: identify normal events
- ▶ **Issues related to analysis**
 - ▶ Possibly inconsistent between misuse and anomaly
 - ▶ Countermeasures and alert context require additional effort
- ▶ **One instead of two is not sufficient**
 - ▶ Not enough forewarning from misuse detection
 - ▶ Not enough context from anomaly detection
 - ▶ Anomaly & misuse detection mixed in unspecified ways
 - Specific « generic » signatures
 - Pre-processing

New sensor architecture



Expected analysis results



▶ False negative rate

- ▶ False negative rate of the anomaly detection module only
- ▶ All events flagged by anomaly detection are suspicious

▶ False positive rate

- ▶ Two factors
 - False positive rate of anomaly detection module
 - False positive rate of misuse detection module
- ▶ $P(\text{normal unidentified event} \approx \text{known malicious event})$
 - Examples: count.cgi, forms, calendar, ...
 - Small ($< 1\%$) number of alerts
 - Small number of snort signatures (20 from 600)
- ▶ Declined in identified attacks and additional workload

▶ Critical: false negative rate of anomaly module



Implementation consequences

▶ Anomaly detection

- ▶ Conservative to limit false negatives
- ▶ Incremental growth
- ▶ High performance

▶ Misuse dedicated to extraction of characteristics

- ▶ References
- ▶ Success
- ▶ Usage (normal, scan, attack)
- ▶ Returned information
- ▶ Pertinence
- ▶ (above order is important)

Cost comparisons



▶ Parameters

- ▶ Percentage of attacks in traffic (0% -> 10%)
- ▶ Cost of each module
- ▶ False negative rate for each module
- ▶ False positive rate for each module

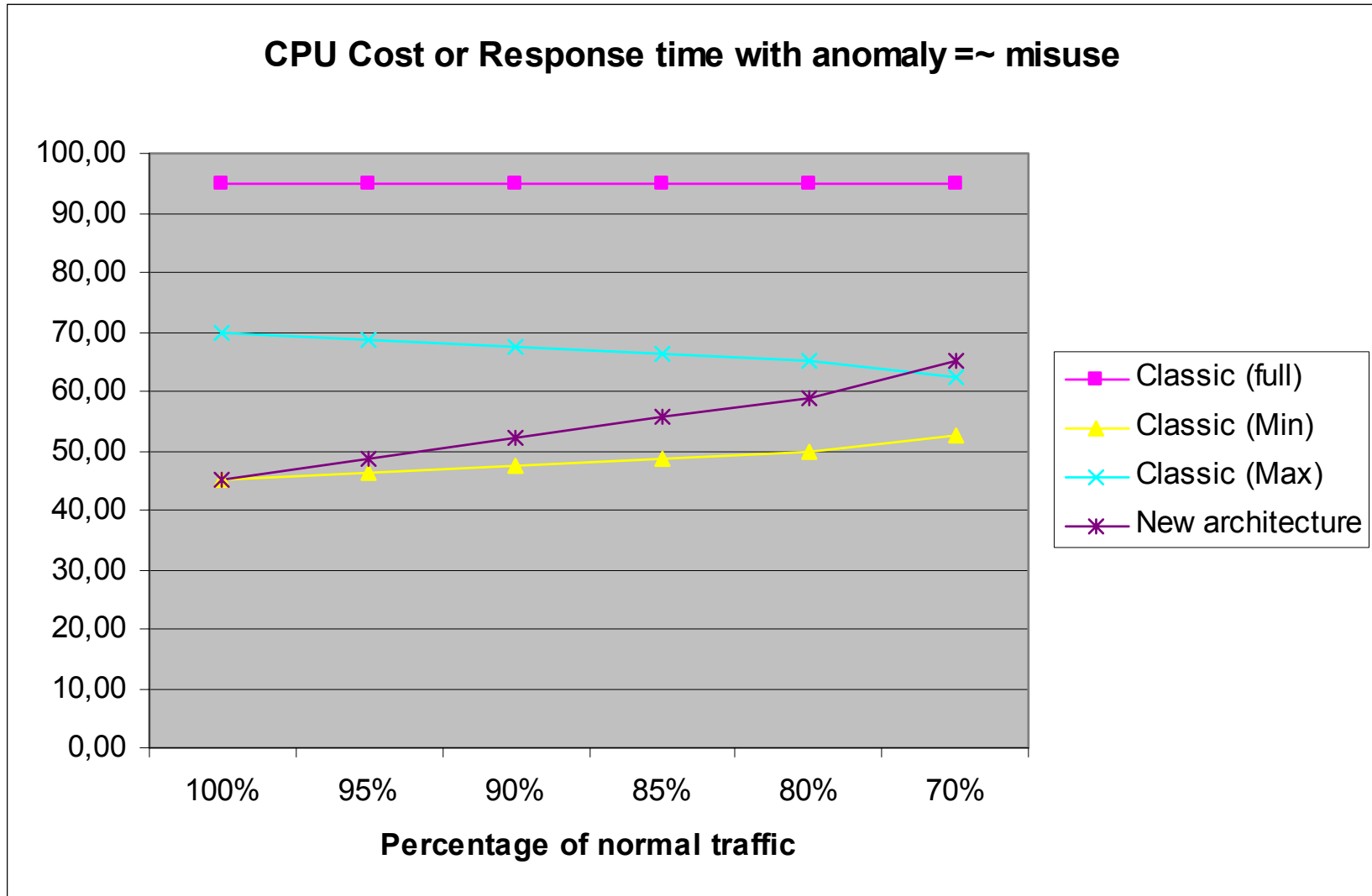
▶ Comparison between the 2 architectures

- ▶ Processing speed
- ▶ Number of unknown alerts to handle

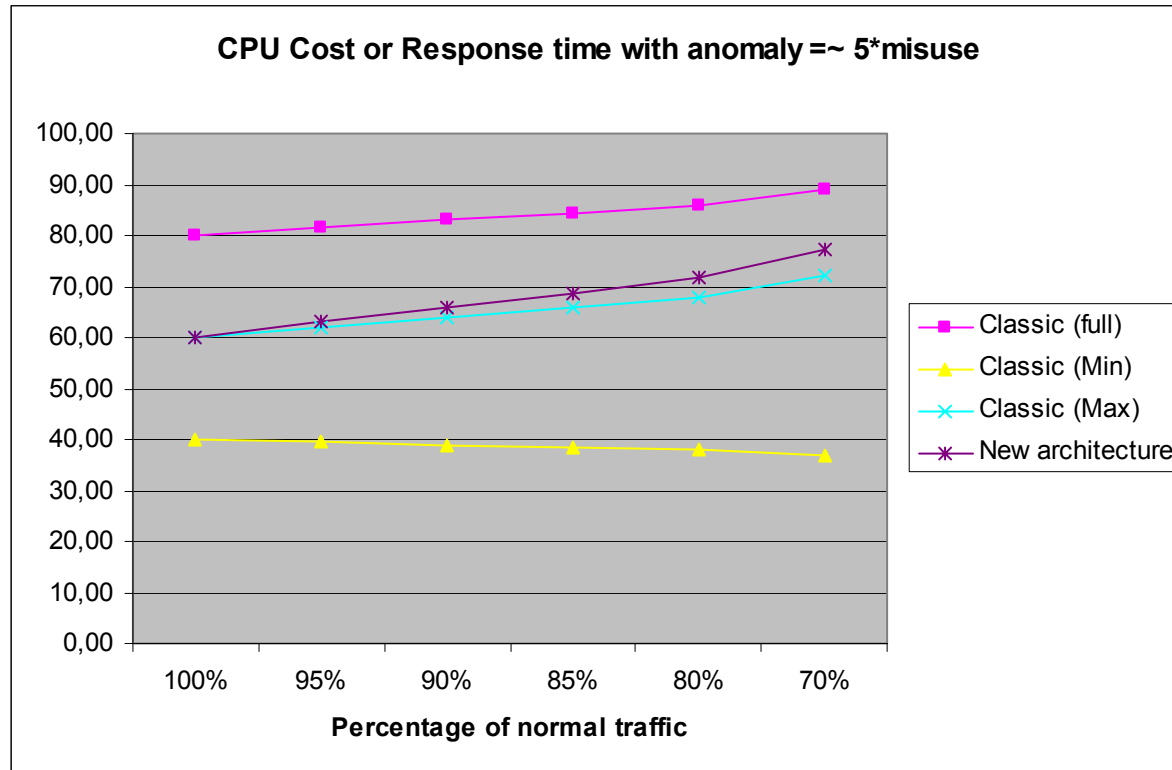
▶ Hypotheses

- ▶ Old arch: 3 possible mixer strategies, always ask, always pick worst, always picks best, false negatives do not occur on same traffic.
- ▶ New arch: filtered traffic has same failure/success distrib

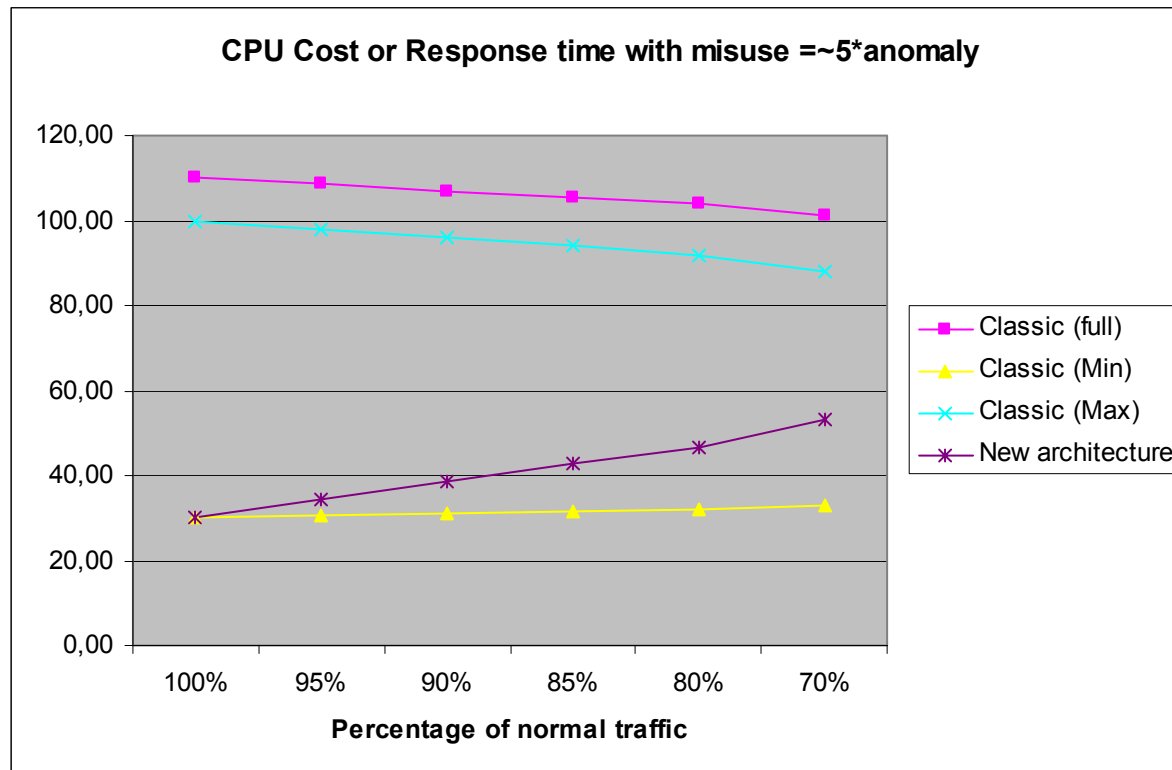
Performance (anomaly \approx misuse)



Performance (anomaly $\approx 5^*$ misuse)



Performance (the opposite)



- ▶ **Advanced IDS sensor architecture**
 - ▶ Focus on false negative rate to limit volume
 - ▶ Use misuse detection to precisely identify issue
 - ▶ Don't look for sensor with worse than 10^{-5} FPR
- ▶ **Upcoming features in supervision**
 - ▶ Sensor management
 - ▶ Signature management
 - ▶ Inventory management
 - ▶ Knowledge exchange
 - Queries database