IBM

IBM Zurich Research Laboratory

Intrusion Detection: Eliminating False Alarms

MAFTIA Final Review

Andreas Wespi

18/19 February 2003 | Newcastle

Overview

- MAFTIA and Intrusion Detection
- Eliminating False Alarms
- Demonstrator of an Intrusion-Tolerant IDS

MAFTIA and Intrusion Detection

Objectives

- We are interested in finding solutions to the well-known problem of the high rate of false positive and false negative alarms generated by current IDSs.

  Intrusion Detection ⇨ MAFTIA

- These false alarms can also be due to attacks against the IDSs themselves, therefore the need to design IDSs which are intrusion-tolerant.

  MAFTIA ⇨ Intrusion Detection

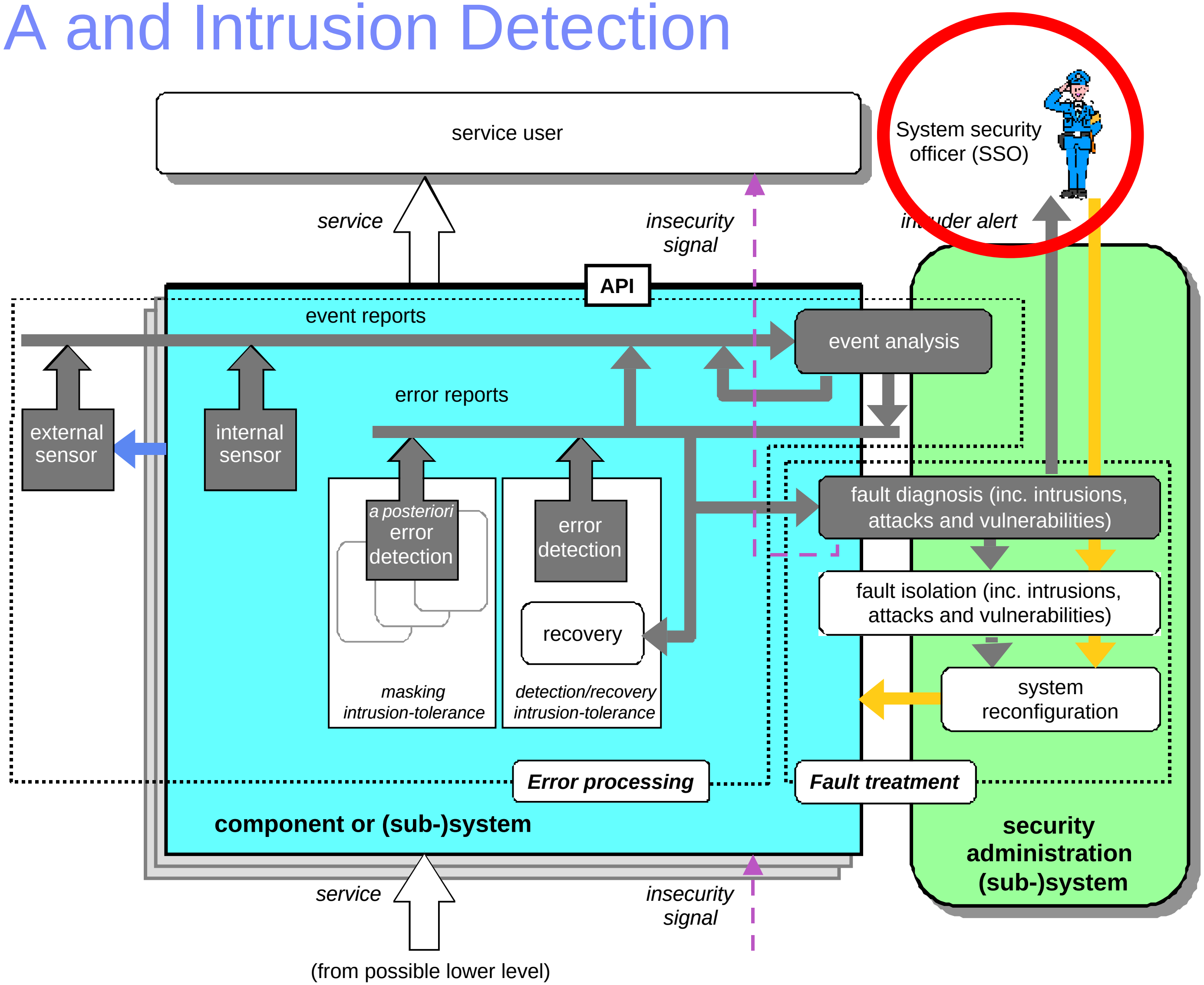
Three main contributions by WP3

- False negatives
  - Method to increase detection coverage by effectively combining IDSs based on a taxonomy, a framework and a tool for analyzing the strengths and weaknesses of IDSs (D3, D10)
- False positives
  - Method and tool to discard false alarms by mining alarm clusters (D10)
- Attacks against the IDS itself
  - Methods and techniques to make an IDS intrusion-tolerant by taking advantage of, among others, techniques developed within other MAFTIA work packages (D10, D13)

Links to other work packages

- Vision: Intrusion detection is a large scale distributed application
- The ultimate goal, a distributed intrusion-tolerant IDS, requires components developed by other MAFTIA work packages, especially WP2 and WP4
- WP3 focuses on the ID issues, a prerequisite to ensure certain assumptions made within other work packages
- WP1 provides the underlying foundations for the taxonomy we have developed to assess the detection capabilities of IDSs

MAFTIA and Intrusion Detection



Eliminating False Alarms

Klaus Julisch

Problem statement

- Intrusion-detection systems (IDSs)
  - monitor and analyze network traffic or event logs,
  - trigger alarms when signs of attacks occur, and
  - have an operator evaluate and respond to alarms.

Sample alarm

10.11.34.2	3319	123	10.11.1.1	80	13:45	http://..cgi?
Source-IP	**Src-Port**	**Alarm-id**	**Dst-IP**	**Dst-Port**	**Time**	**Context**

- Practical problems of IDSs include:
  - 1000s of alarms per day and false alarm rates above 95%;

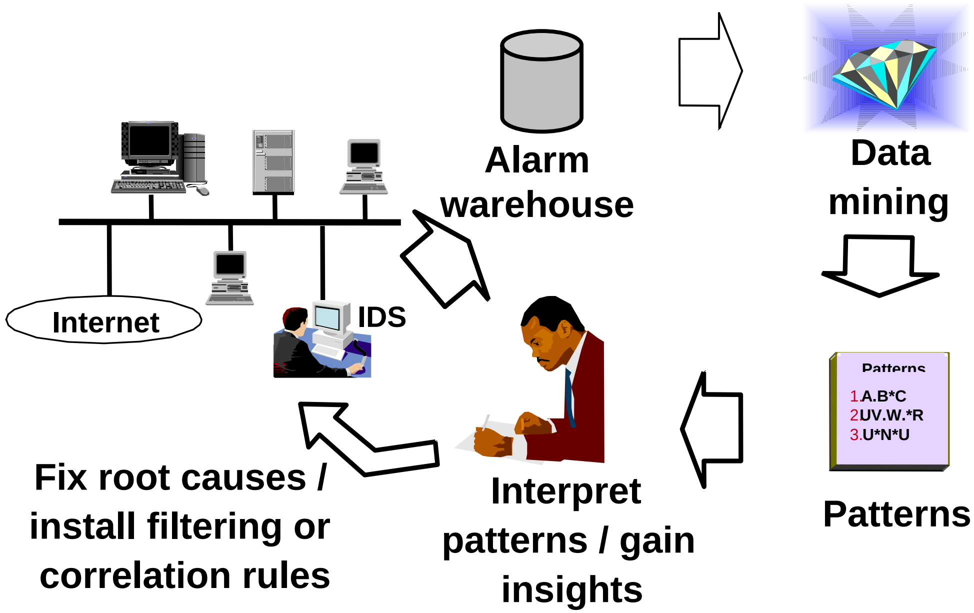
- Corollaries:
  - Manually finding the true attacks is expensive & error prone.
  - Tools are needed that automate alarm evaluation!

A similar problem statement …

How to find a needle in a haystack?

Our approach

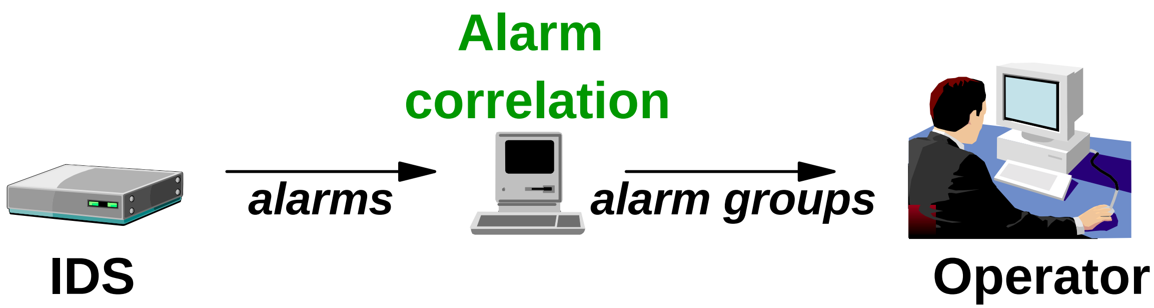
- Idea: Learn from the past to master the future.
- Mine patterns from historical IDS alarms.
- Understand why patterns occurred (i.e. identify alarm root causes).
- Act to reduce the future alarm load (e.g. fix, block, filter, ...).



Note 1: Patterns are assumed to be *persistent*.

Note 2: Mining is *off-line* and uses historical alarms!

Comparison to the “classic" alarm handling approach



Alarm correlation tries to group alarms that pertain to the same phenomenon.

- Alarm correlation versus mining of actionable knowledge
- Real-time (& resource constraint) versus off-line.
- Reconstructing attack scenarios versus finding persistent and mostly benign patterns.
- Generic versus custom-made (i.e. site- & IDS specific).

Data mining technique requirements

- To be of value in our framework, a prospective data mining (DM) technique should be:
  - Scalable to up to several millions of alarms.
  - Noise tolerant (alarms can come "out of the blue").
  - Multi attribute type (categorical, numerical, time, string, ...).
  - Easy to use for non-DM experts.
  - Yielding interpretable & relevant patterns.

- Key question: Which DM techniques qualify?

- We investigate the suitability of episode rules and conceptual clustering.

Episode rules

- Episode rules predict the occurrence of certain alarms based on the occurrence of other alarms:

  $\langle A_1, \dots, A_k \rangle \Rightarrow \langle A_{k+1}, \dots, A_n \rangle$ [supp, conf, window-width]

- Useful episode rules we found:
  - Attack tools, IDS idiosyncrasies, system administration tasks

- Problems encountered:
  - Abundance of episode rules generated.
  - Difficulty to interpret episode rules.
  - Weak predictiveness, alarm reduction of only about 1 %.

- Conclusion: Episode rules do not solve our problem.

Conceptual clustering

- Clustering organizes a data set into groups of similar objects.
- In conceptual clustering, objects are similar iff they possess a "simple" intentional description in some language (e.g. predicate calculus).
- Example cluster:

Name	Sex	Age	Job
Bob	male	41	chemist
Jim	male	57	physicist
Chuck	male	44	astronomer
Ted	male	52	chemist

Advantages: ***Understandability*** **and** ***support for categorical attributes.***

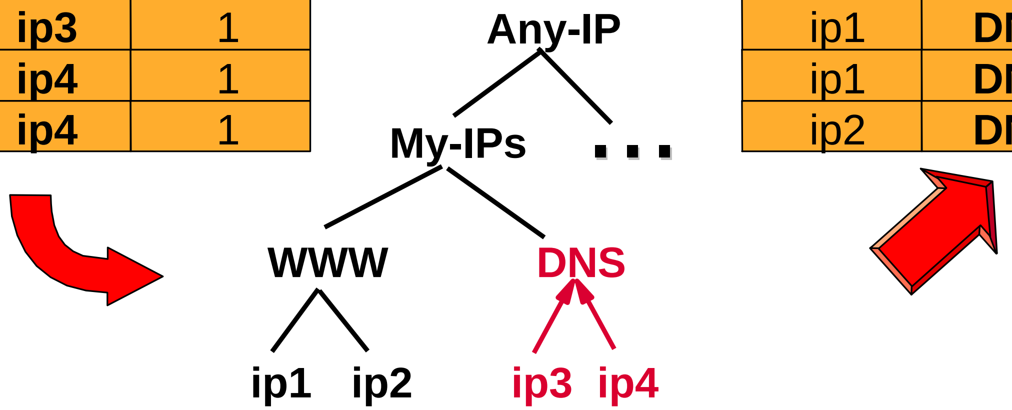
Intentional description:
$\{x \mid (x \text{ is male}) \wedge (x \text{ is over } 40) \wedge (x \text{ is scientist})\}$

Attribute-Oriented Induction (AOI)

- Attribute-Oriented Induction (AOI)
  - summarizes relational database tables
  - by repetitively replacing attribute values by more abstract ones,
  - which are taken from user-defined generalization hierarchies.

Original table

Src-IP	Dest-IP	Count
ip1	**ip3**	1
ip1	**ip4**	1
ip2	**ip4**	1



After generalization step

Src-IP	Dest-IP	Count
ip1	**DNS**	1
ip1	**DNS**	1
ip2	**DNS**	1

Attribute-Oriented Induction (AOI)

- Merge identical tuples and update their counts:

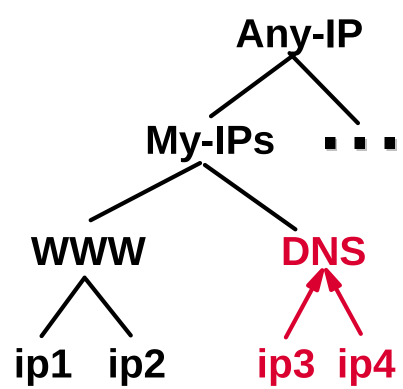
After generalization step

Src-IP	Dest-IP	Count
ip1	**DNS**	**1**
ip1	**DNS**	**1**
ip2	DNS	1

After merging

Src-IP	Dest-IP	Count
ip1	**DNS**	**2**
ip2	DNS	1

- AOI terminates when each attribute assumes at most d_i distinct values.
  - The d_i, i=1, ..., n, are user-provided.
  - Example: Be $d_{Src\text{-}IP} = d_{Dest\text{-}IP} = 1$ then terminate after generalizing Src-IP. Final result: (WWW, DNS, 3)



Algorithm

Input : Alarm table *T,* Hierarchies *H[i*];

Output: Alarm clusters represented by generalized alarms;

1. **for all** alarms *a* in *T* do *a.C := 1;* *// Init counts*
2. **while** table *T* is not abstract enough **do** {
3. Select an alarm attribute *A[i];*
4. **for all** alarms *a* in *T* **do** *// Generalize A[i]*
5. *a.A[i] :=* parent of *a.A[i]* in *H[i];*
6. **while** identical alarms *a, a'* exist **do** *// Merge*
7. Set *a.C := a.C + a'.C* and delete *a'* from *T*;
8. }

Applying AOI to IDS alarms

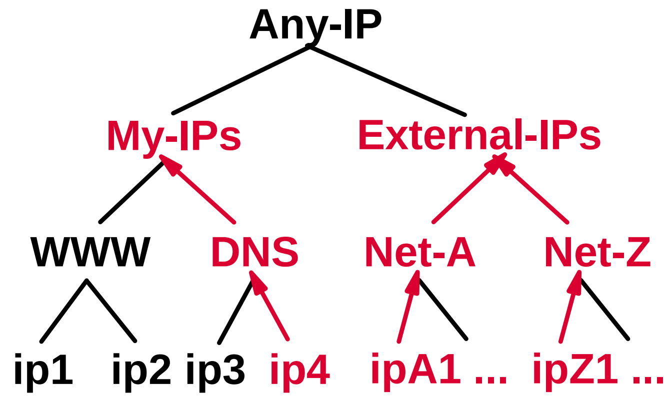
- Problem: IDS alarms are skewed and noisy, which leads to over-generalization!

Typical IDS alarm log

Src-IP	Dest-IP	Count
ip1	ip4	1000
ip1	ipA1	1
ip1	ipB1	1
...	...	...
ip1	ipZ1	1

main signal (row ip1 / ip4 / 1000)

26 noise alarms (rows ip1 / ipA1 through ip1 / ipZ1)



Dest-IP is generalized twice if $d_{Dest\text{-}IP} < 27$:

ip1	My-IPs	1000
ip1	External-IPs	26

Over-generalized! (circled row: ip1 / My-IPs / 1000)

Modifications to “classic” AOI

- Modification 1:
  - Abandon the d_i thresholds for the number of different attribute values.
  - Introduce min_size $\in$ N, and generalize until a cluster C has a count bigger than min_size (i.e. C.count > min_size).
- Modification 2:
  - Remove clusters of a size larger than min_size and
  - reset the remaining alarms (i.e. undo all generalization steps).
- Modification 3:
  - Use heuristics to select a suitable attribute for generalization.

Sample result of modified AOI

Concepts like "External-IPs" come from the hierarchies!

Cluster of size 50573 alarms:

Alarm Name:	IIS View ASP Source Attack
SrcIP/SrcPort:	External-IPs / Non-Privileged
DstIP/DstPort:	10.8.17.* / 80
Context:	Get /cgi/s?action=...www%2Euva%2E...
Time Structure:	weekdays

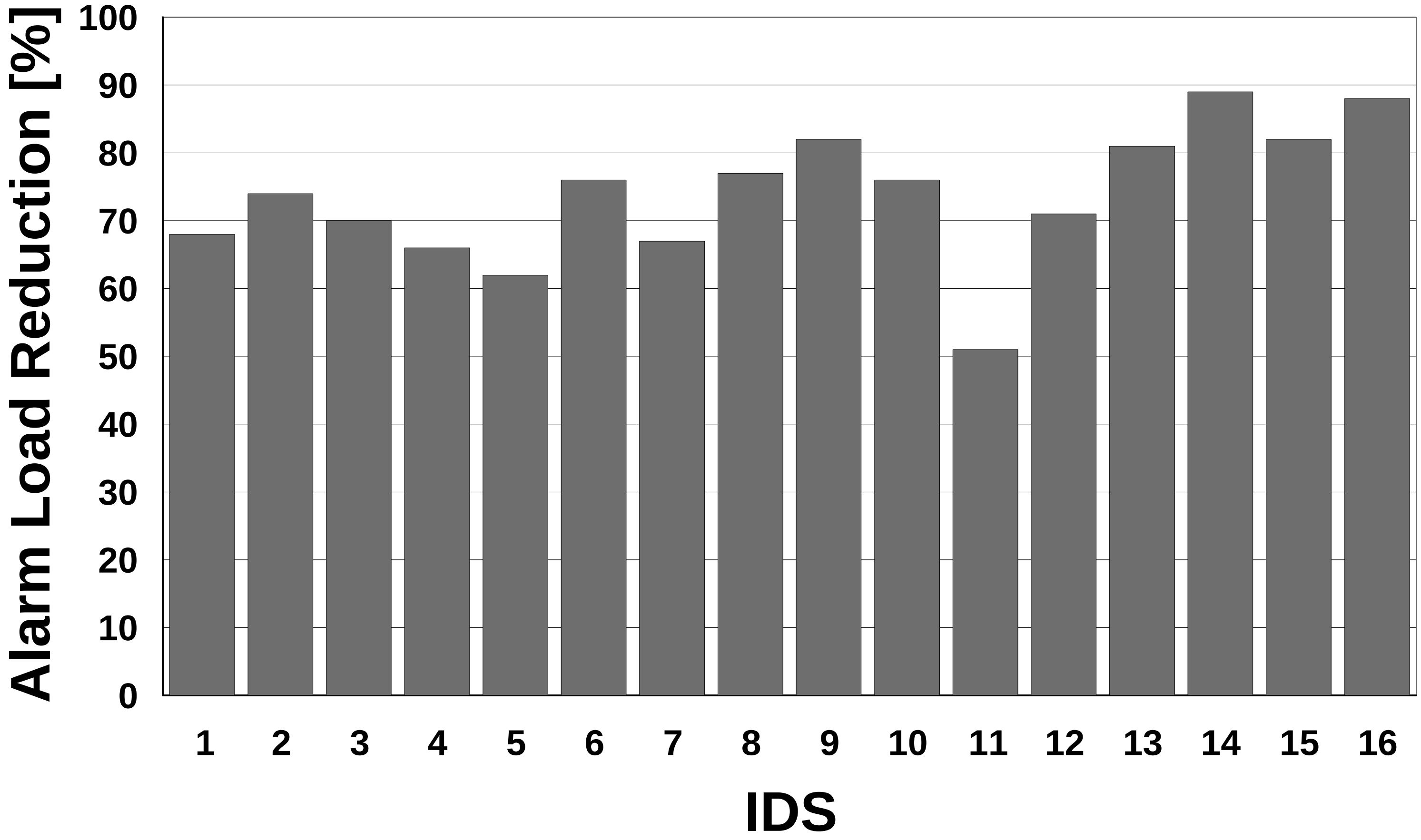
Experimental evaluation

- All experiments use real-world data.

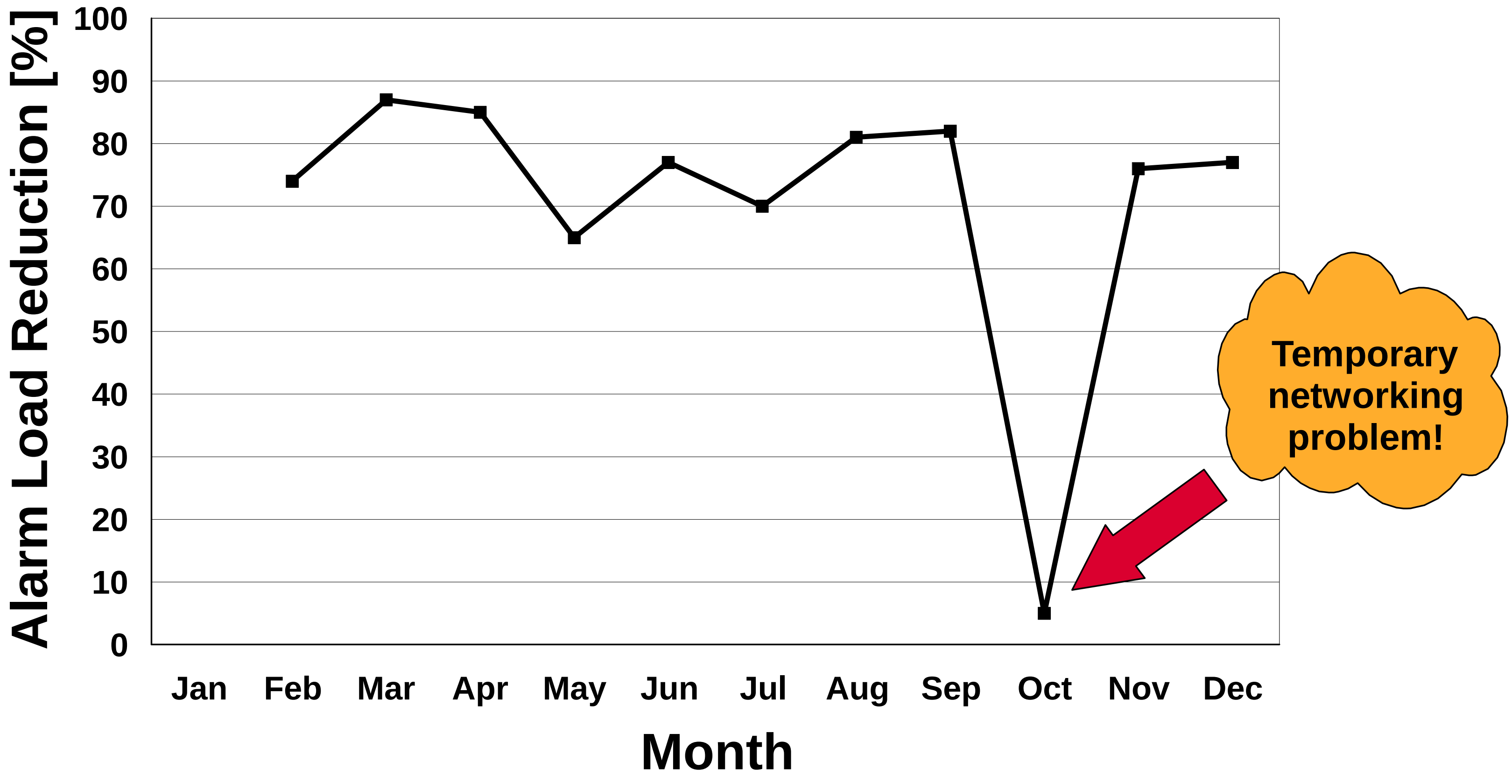
- Experimental setup:
  - Choose an IDS X and month m.
  - Cluster the alarms that IDS X triggered in month m.
  - Interpret the clusters obtained.
  - Manually derive filtering rules for the alarms in the clusters.
  - Evaluate the filtering rules on the alarms of IDS X in month m+1.
  - Measure the alarm load reduction.

- Two sets of experiments:
  - Fixed month, varying IDS.
  - Fixed IDS, varying month.

Alarm load reduction per IDS in December 2001



Alarm load reduction over the year 2001 for IDS 8



Summary

- Problem: IDSs tend to generate a flood of mostly false alarms.
- Suggested solution: Mine, understand, act.
  - Episode rules: Expensive to use, minor alarm reduction.
  - Conceptual clustering: Well-suited in our framework.

- More abstractly, we have shown that:
  - IDS alarms have a pronounced and persistent structure.
  - Data mining can be used to reveal this structure.
  - Knowledge of this structure is actionable, i.e., can be used to handle future alarms more efficiently.

Dissemination

- Publications at RAID 2000/2001, ACSAC 2001, ACM CCS 2001, KDD 2002
- Technology deployed by IBM's Managed Security Services organization
- In plan for next release of IBM Tivoli Risk Manager