

MONAM 2006

Leveraging multiple approaches in intrusion detection – towards unified detection

Hervé DEBAR
France Télécom Division R&D

September 29th, 2006



research & development



Agenda

- Introduction and motivations
- Combination of multiple detection algorithms
- Experimentation results
- Unified detection

Quick introduction to intrusion detection

- Analyze activity occurring on an information system

- Activity captured from (input information)
 - Network packets
 - System events
 - Application logs

- Analysis strategy
 - Misuse detection
 - Use knowledge of vulnerabilities and attacks
 - Trigger alert when evidence found
 - Anomaly detection
 - Defines appropriate behavior
 - Trigger alert when model mismatches

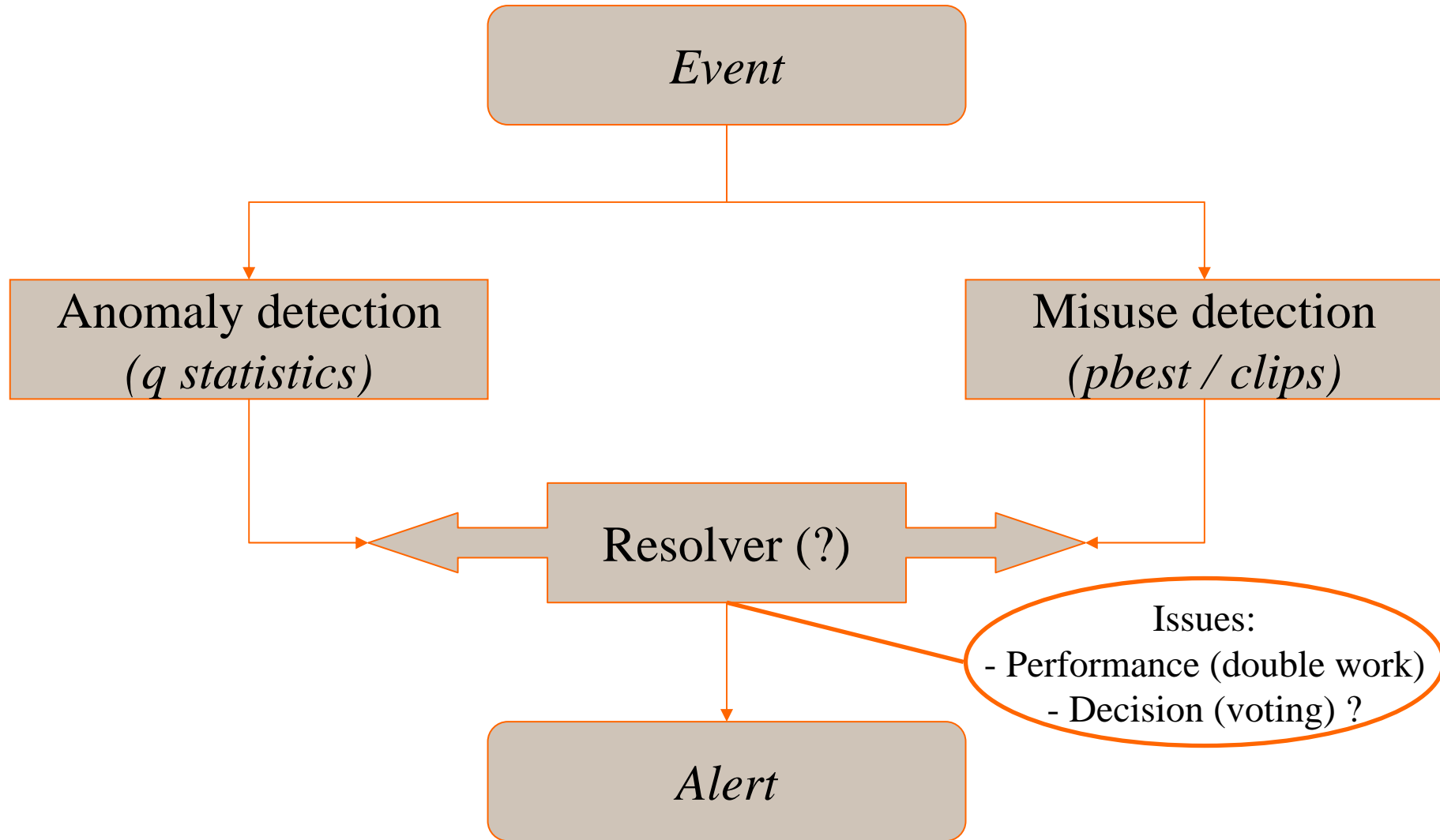
Detection problems

- Anomaly detection
 - Understand alerts and react to them
 - New user behaviours

- Misuse detection
 - Express the appropriate knowledge
 - New attack behaviours (e.g. new vulnerabilities)

- Neither approach identifies **new** usages or security issues.
 - Combination may improve coverage.

Parallel combination (NIDES 1988)



Back to basics

Anomaly Detection

False negatives	False positives
Normal (known)	Attack

Misuse Detection

False negatives	False positives
Normal	Attack (known)

3-states diagnosis

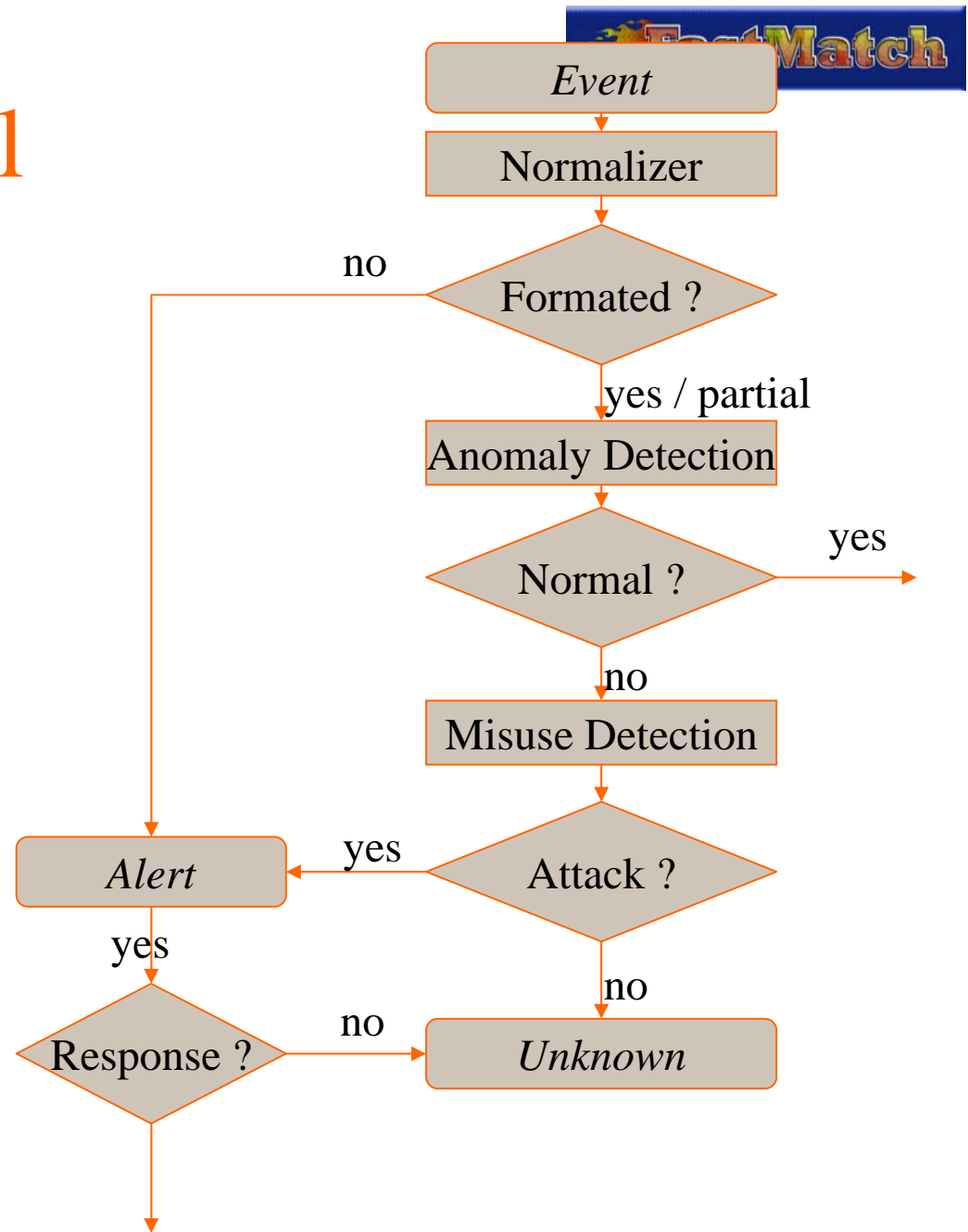


Agenda

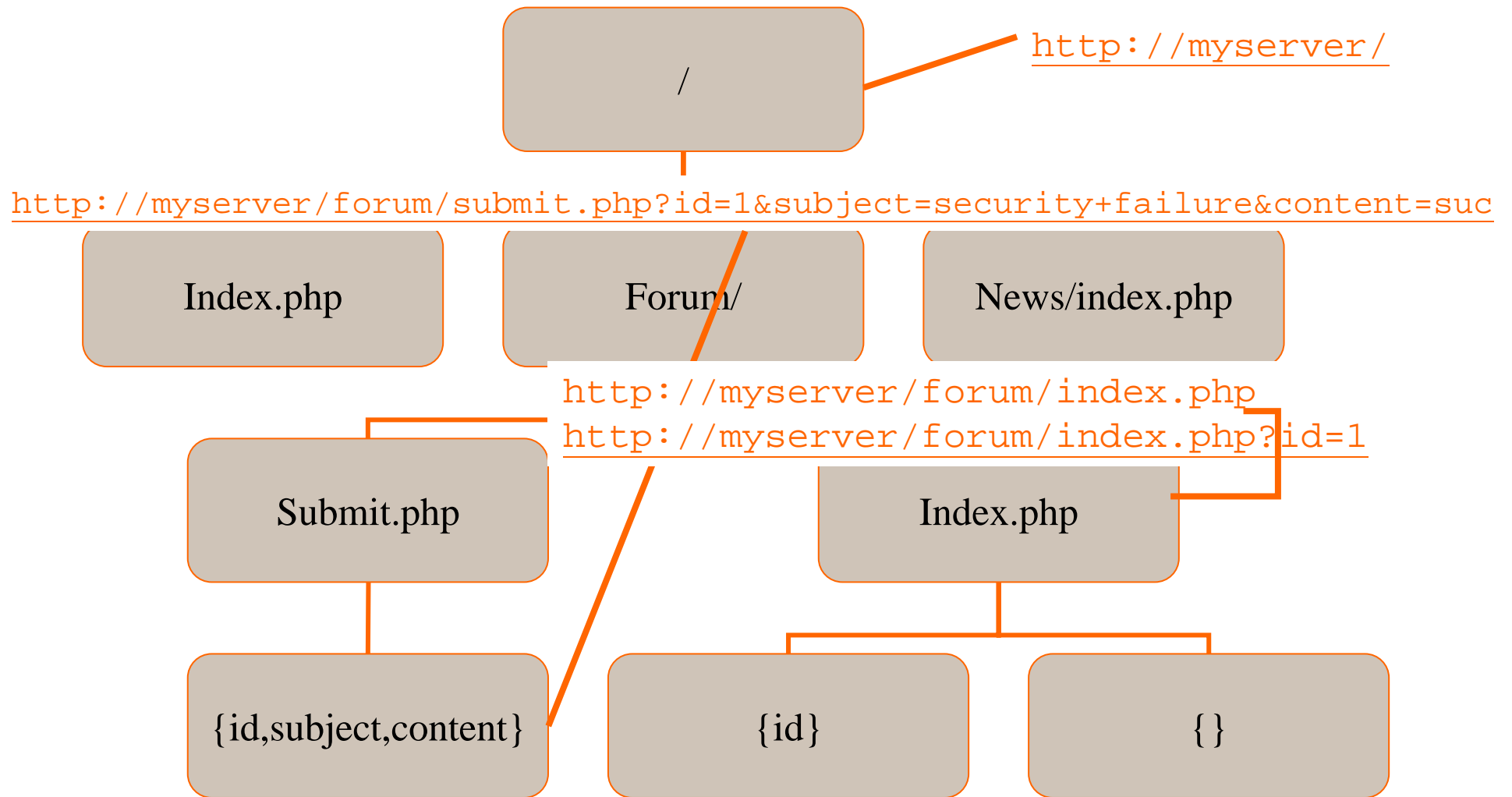
- Introduction and motivations
- Combination of multiple detection algorithms
 - Anomaly detection
 - Misuse detection
- Experimentation results
- Unified detection

Cascade proposal

- Proposal: carry out one analysis on the residuals of another one
 - Which one to do first ?
 - How to justify it ?
 - How to manage scores / rankings ?
- Misuse knowledge update issues
 - Timely update
 - Graduated diagnosis
- Proposal: anomaly first
 - Potentially faster
 - Potentially less false negatives
 - Misuses better qualifies input
 - Specificities of our misuse detector (ranking)
 - Specificities of our anomaly detector (clear-cut decision)



Simple anomaly detector (resource tree)



Characteristics of resources

- Eliminated fields
 - IP address
 - Size
- Fields used for characterizing resources
 - Existence of auth data (not the data itself)
 - Protected resource
 - Timestamp (week-end, week-day)
 - Method (GET, POST, HEAD, anything else)
 - Existence of parameters (dynamic resource)
 - Protocol (1.x or 0.9)
 - Response (status code)
- Additional computed variables (volume information)
 - Average number of requests per day
 - Proportion of this request among the others per day

Clustering

Group	Nb of resources	Percentage	Number of requests	Percentage
1	215	0,99%	1051	0,12%
2	12751	58,82%	714115	82,46%
3	2216	10,22%	74981	8,66%
4	4483	20,68%	10014	1,16%
5	1628	7,51%	1965	0,23%
6	386	1,78%	63911	7,38%

Group interpretation

- Group 2: successful GET requests (200, 300)
 - Normal activity of web server
- Group 6: redirected GET requests (300)
 - Small in individuals, large in requests
 - Also representative of normal activity
- Group 3: unsuccessful GET and HEAD
- Group 4: similar to 3 but focusing on day-of-week
- Group 5: similar to 3 but focusing on week-end
- Group 1: important variance on all variables

Group profiles summary

Profile	Name	Groups
Method + status code	Successful GET	2,6
	Failed GET	3,4,5
	Trash can ...	1
Request by day	All days	2,3,6
	Separation WD/WE	1,4,5
Volume	Large	2
	Average	3,6
	Small	1,4,5

Model of normal behaviour

- Group 2 + 6: **normal**
 - 90% of activity on well defined resources
- Group 4 + 5: **not normal**
 - 28% of resources for only 2% of requests
 - No particular issue as well
- Group 3
 - Close to 2 and 6, but on 404
 - Interpretation: recurrent errors on automated processes
 - Can also be demonstrative of failed worm attempts
 - Choose to integrate into normal for the moment
- Group 1
 - Too much statistical variation for assignment into model

Model evaluation

Group	In model	Number of resources	Malicious resources
1	No	216	23
2	Yes	12751	0
3	Yes	2219	24
4	No	4483	111
5	No	1628	386
6	Yes	386	0

- It is possible to construct a simple behaviour model
- Missing a few **failed** attempts

Agenda

- Introduction and motivations
- Combination of multiple detection algorithms
 - Anomaly detection
 - Misuse detection
- Experimentation results
- Unified detection



Misuse detector

■ 3-step process

- Normalization (regexps)
 - Clean and segment input
 - Create derived fields
- Feature extraction (regexps)
 - Organized search
- Feature correlation (prolog)

■ Unique signature language

- Vulnerability usage
- Evasive actions
- Attack template patterns
- Sensitive system targets

■ Continuous evaluation of risk

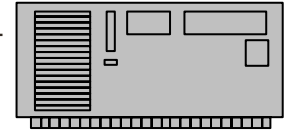
```
<signature name="phf"
  trigger="/phf$"
  severity="+1"
  class="query, apache, cgi" >
</signature>
<description origin="cve">
  <name>CVE-1999-0067</name>
  <url>http://cve.mitre.org/cgi-bin/cvename
</description>
<description origin="bugtraqid">
  <name>629</name>
  <url>http://www.securityfocus.com/bid/6
</description>
</signature>
```

```
<signature name="success_cgi"
  trigger="pattern(status_200),
  class(cgi)"
  severity="+3"
  class="rule">
</signature>
<description origin="vendor-specific">
  <name>If a CGI script referenced as dan
  status, then the severity is increased.
  <url>file://./signatures.xml</url>
</description>
</signature>
```

Misuse analysis (Successful attack)



http://cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd



200 OK

1.1.1.8 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0" 200 2450

Severity :

non_ascii

12

1

C0: severity <= 0, no attack

C1: severity in [1,4], suspicious evasive activity

C2: severity in [5,7], incomplete diagnosis

C3: severity in [8, .. [, possibly successful compromise

status_200

0

get_method

0

cgi_dir

0

phf (cgi_dir)

1

(implies cgi)

etc_password

1

(implies file)

args_not_empty

0

unix_cmd

1

real_attempt

2

(cgi + file)

success_cgi

+3

(cgi + status_200)

success_file

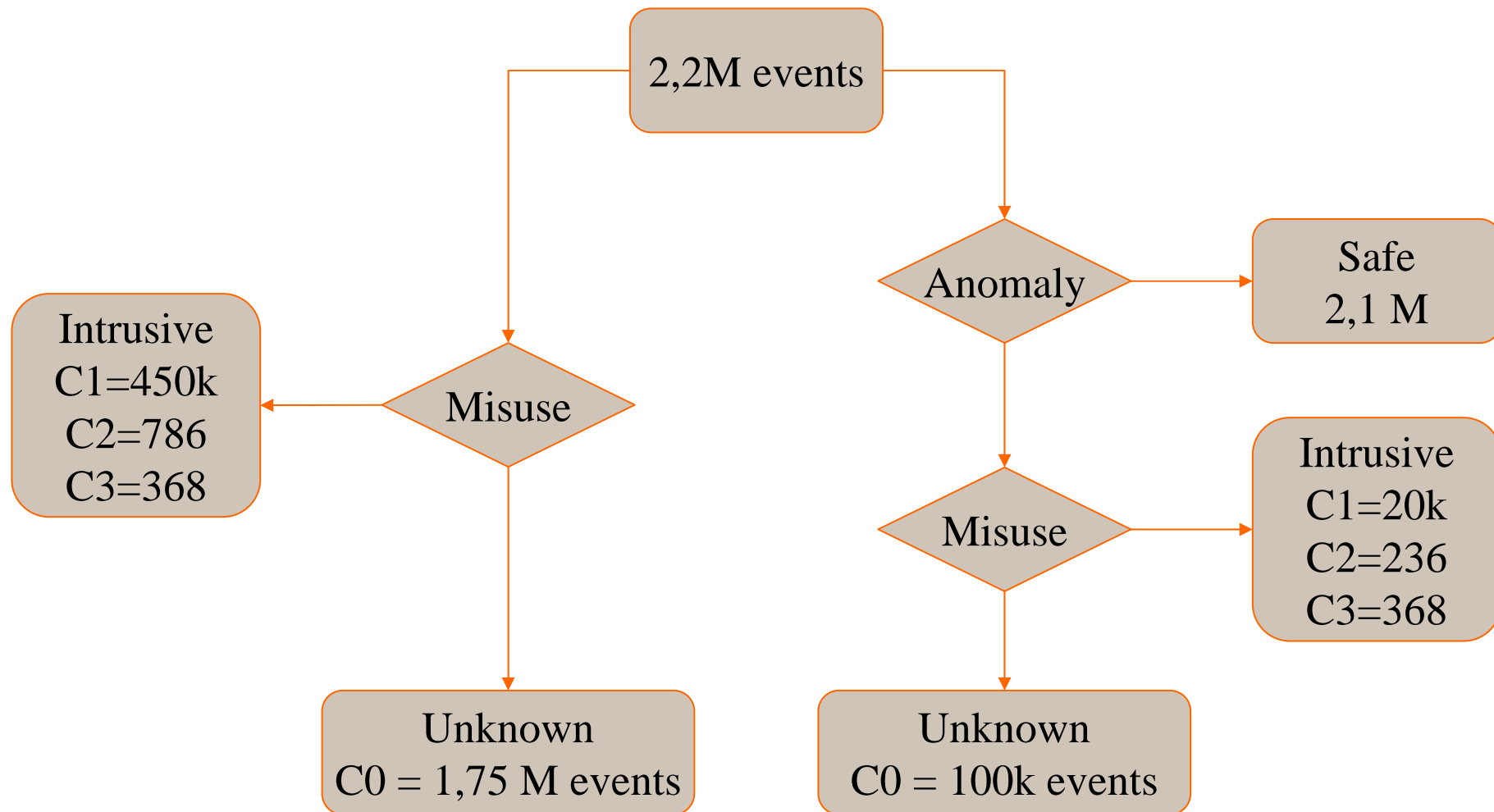
+3

(file + status_200)

Agenda

- Introduction and motivations
- Combination of multiple detection algorithms
- Experimentation results
- Unified detection

Example results



Manual analysis of the combination results

- Safe events (2.1M)
 - No attack found

- Intrusive events (20k)
 - C1 : False positives remains
 - C2 : Most false positives eliminated
 - C3 : Real attacks

- Unknown events (100k)
 - No attack found

- Note: false positive = no operator action required

Falses positives & false negatives

- False positives: sensitivity to anomaly detection
 - General intrusion detection issue
 - In this particular case: biased towards **failed** attacks
 - Carry-over of the deviations downstream
 - Must have a **better** detector
- Unknown and change are within the detection process
 - Need to support the security operator
 - Provide mechanism to update models (no on-line learning)
- Qualification enables response to events
 - More accurate (targeted)
 - Automated (less errors)
 - Includes **response** (transaction level)

Agenda

- Introduction and motivations
- Combination of multiple detection algorithms
- Experimentation results
- Unified detection

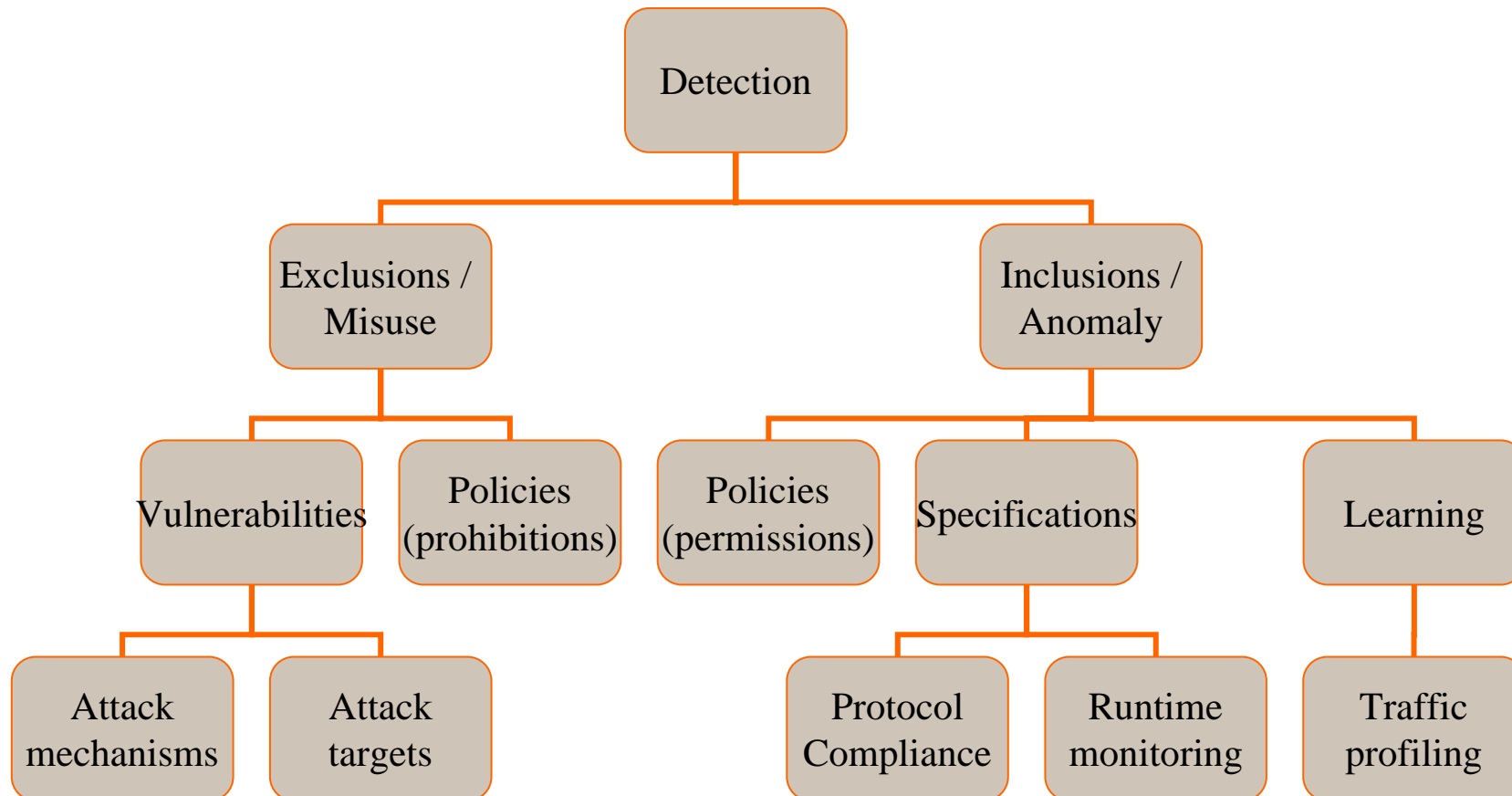
Evolutions of operational security

- Misuse detection is becoming more complicated
 - Increased volume of knowledge
 - Increased number of possibilities to evade detection
 - Polymorphism is becoming part of any attack vector
 - Rootkits almost invisible through virtualization
- Firewalls are being transparently traversed
 - Client-side attacks such as browser vulnerabilities
 - Transparent applications (e.g. Skype)
 - Externalized applications (e.g. Webex)
- Need to re-think security policies
 - Anomaly detection might be needed for stealthy protocol detection
 - More user-oriented responses might be needed to accommodate grey detection

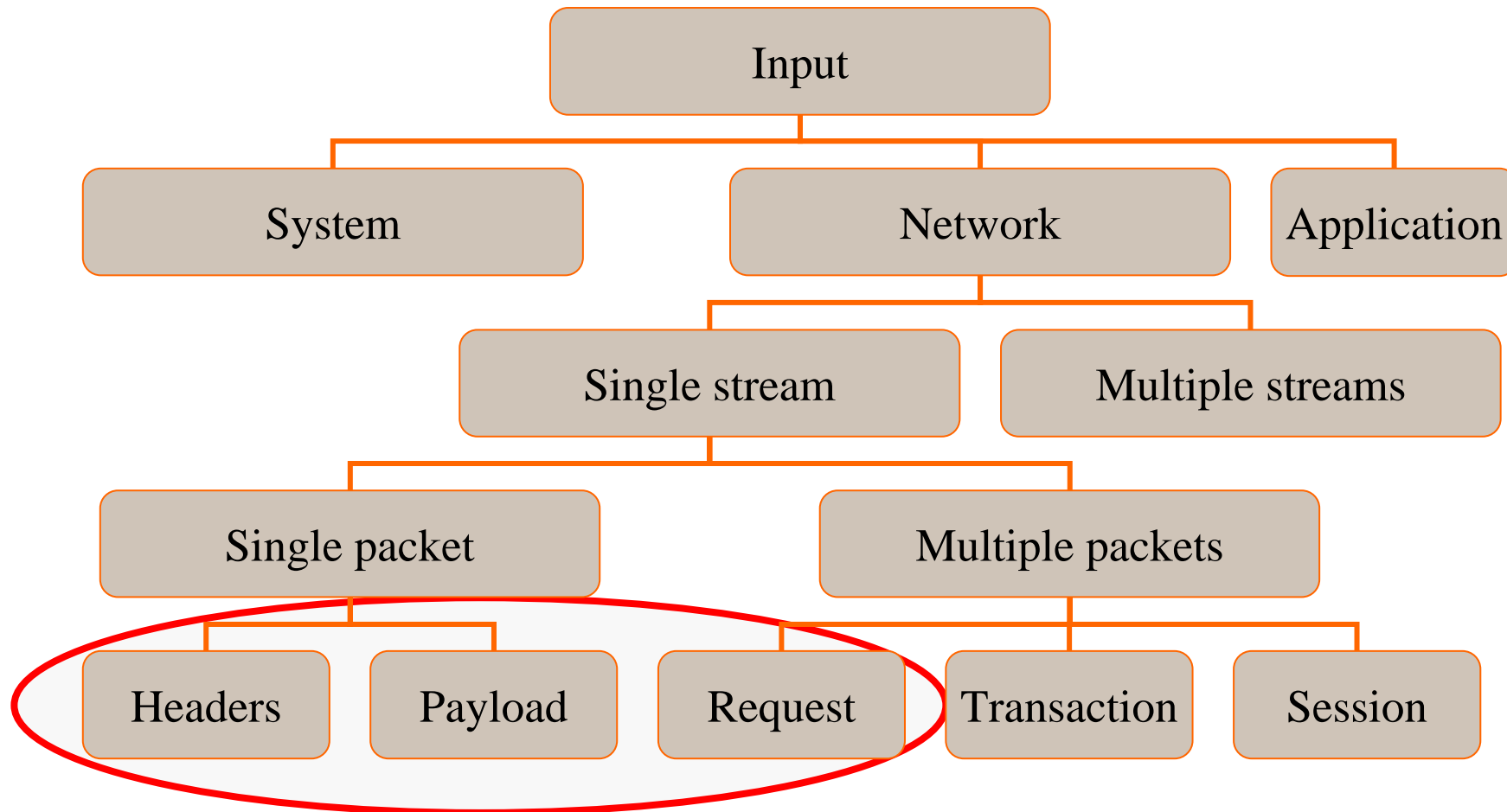
Basic security functions

- Management
 - Description of the "security policy"
 - Segmentation and compilation
 - Broadcast and installation
 - Test and validation
- Evaluation / Detection
 - Receive an incoming request
 - Evaluate its content
 - Request additional information
 - Provide decision
- Application / Reaction
 - Respond to attacks
- Decision / Correlation
 - Bridge between detection and reaction
 - Includes more complex processes (external information, longer-term observation)

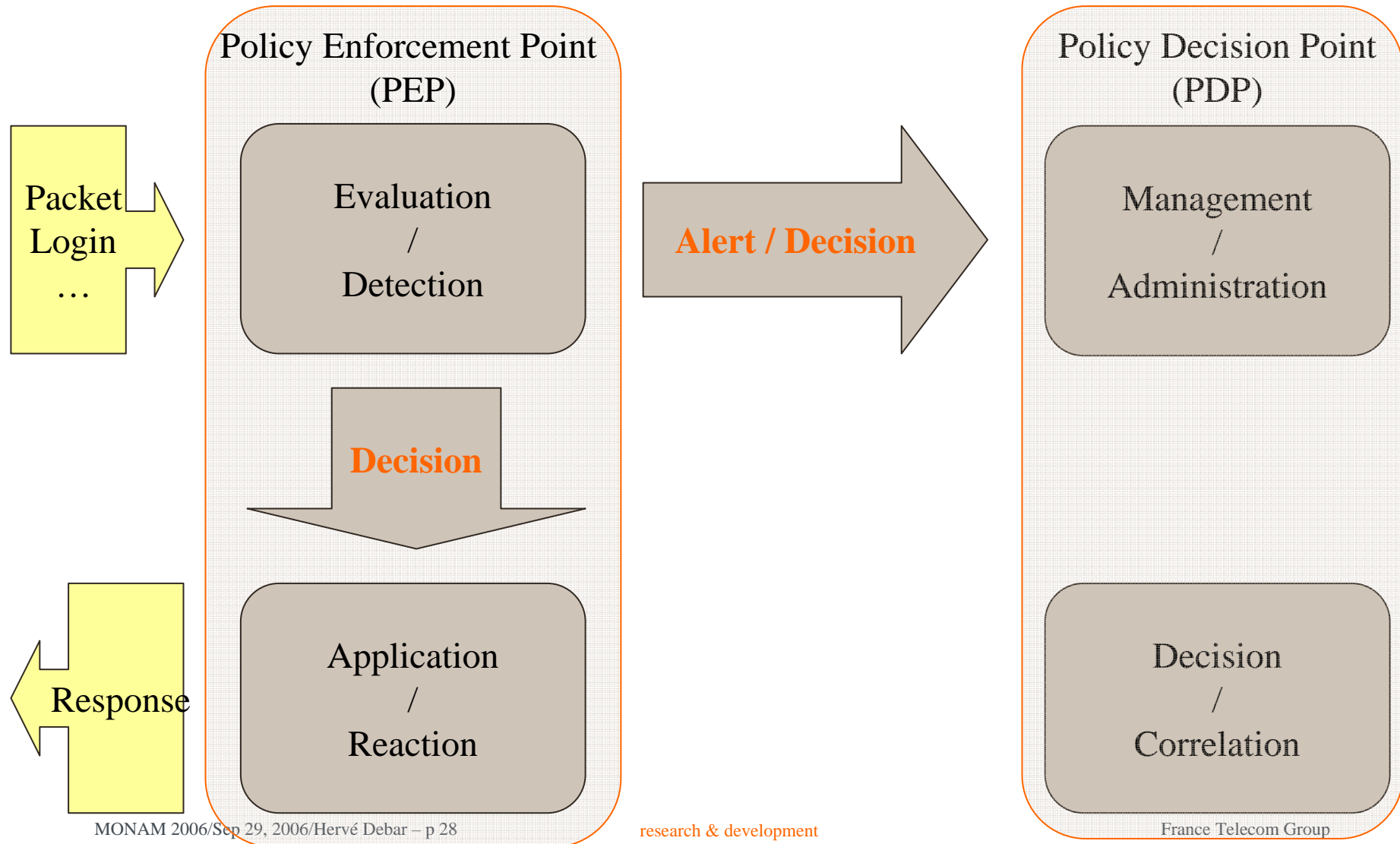
Detection algorithms feature map



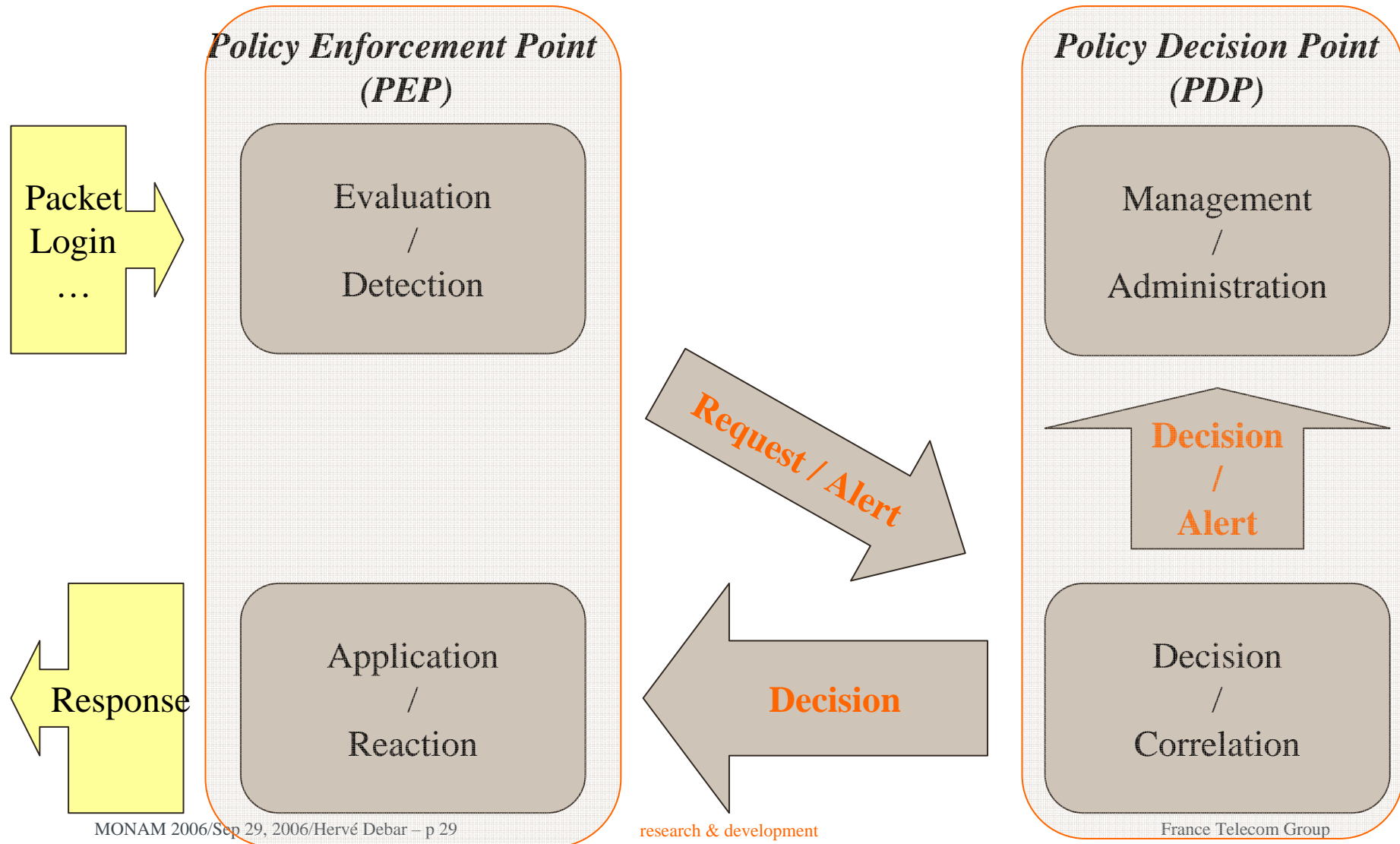
Input feature map



Domain functions (simple input)



Domain functions (complex input)



Future directions

- Separate functions implies bridging
 - Subject to performance constraints
 - Unified configuration language (regexps, stats, inference, ...)
- Rethink what reaction / response is about
 - Need more than accept or block
 - Convenience and visibility of the response to the user and the security officer
 - Duration and graduation of response
 - Failsafe mechanisms to ensure quality of service, service agreements and legal constraints
- Management of grey diagnosis
 - Support for understanding why the input was qualified as grey
 - Support for updating the decision and response configurations