FOSAD 2005

france telecom

Security of operational systems

Intrusion detection

Hervé DEBAR, France Télécom R&D

The present document contains information that remains the property of France Telecom. The recipient's acceptance of this document implies his or her acknowledgement of the confidential nature of its contents and his or her obligation not to reproduce, transmit to a third party, disclose or use for commercial purposes any of its contents whatsoever without France Telecom's prior written agreement.

(Unrestricted)

The 6 dumbest ideas in computer security (Marcus J. Ranum)

- **Default Permit**
  - Enumerate what you accept rather than what you refuse
- **Enumerating Badness**
  - Badness outnumbers goodness and always evolves
- **Penetrate and Patch**
  - Anything you have to change periodically does not work
- **Hacking is Cool**
  - Hacking is a criminal activity and a social problem
- **Educating Users**
  - If it really worked it would have worked by now
- **Action is Better Than Inaction**
  - Beware of new ideas and spending on new magic technology

From: http://www.ranum.com/security/computer_security/editorials/dumb/

Agenda

- **Vulnerabilities**
- **Intrusion detection**
  - Misuse detection
  - Anomaly Detection (new magic word)
    - Behaviour-based
    - Heuristics (anti-virus)
    - 0-day
- **Alert representation**
  - IDMEF
- **Alert correlation**
  - Logical correlation
  - Statistical correlation

Security Information Management

(Unrestricted)

Security techniques classified as

- **Prevention of security issues**
  - User identification and authentication
  - Traffic filtering
  - Encryption of communication streams or data
- **Detection of security incidents**
  - Specific intrusion detection sensors
  - Gathering and analysis of execution traces
- **Recovery from security incidents**
  - Backup and restore procedures
  - Business continuity plans
  - Confidentiality failures are usually irrecoverable
- **Properties determined by the security policy**
  - According to cost and business objective constraints

Intrusion prevention
is mostly marketing because
-Already existed in IDS sensors
-Only applicable to a few alerts

(Unrestricted)

We now know that

- **All these mechanisms can fail**
  - Detection mechanisms as well
- **In known or unexpected ways**
  - Most often with unexpected consequences
- **Usually with some warning before failure**
- **That is what "operational security" is about**
  - Live with possible failure

(Unrestricted)

Operational security

- **Networks**
  - Core (IP backbone)
  - Access (DSLAM, BAS, …)
  - Underlying infrastructure services (DNS, Radius, DHCP, …)
- **Services**
  - Information system infrastructure
  - Information

(Unrestricted)

Vulnerabilities

Basic conception flaws

- IP Spoofing
- TCP session highjacking
- DNS highjacking
- DNS cache poisoning

Design flaws

- PPTP
- SSL v1
- Passwords
- HTTP authentication
- HTTP cookies

(Unrestricted)

Vulnerabilities (2)

Implementation flaws

- Input sanitation (Cross-Site scripting, web programming, …)
- Buffer overflows (bad programming practices)
- Format string vulnerabilities (specification abuse)
- Race condition vulnerabilities (non-atomic transactions)
- Random number generation (seeding error, predictability)

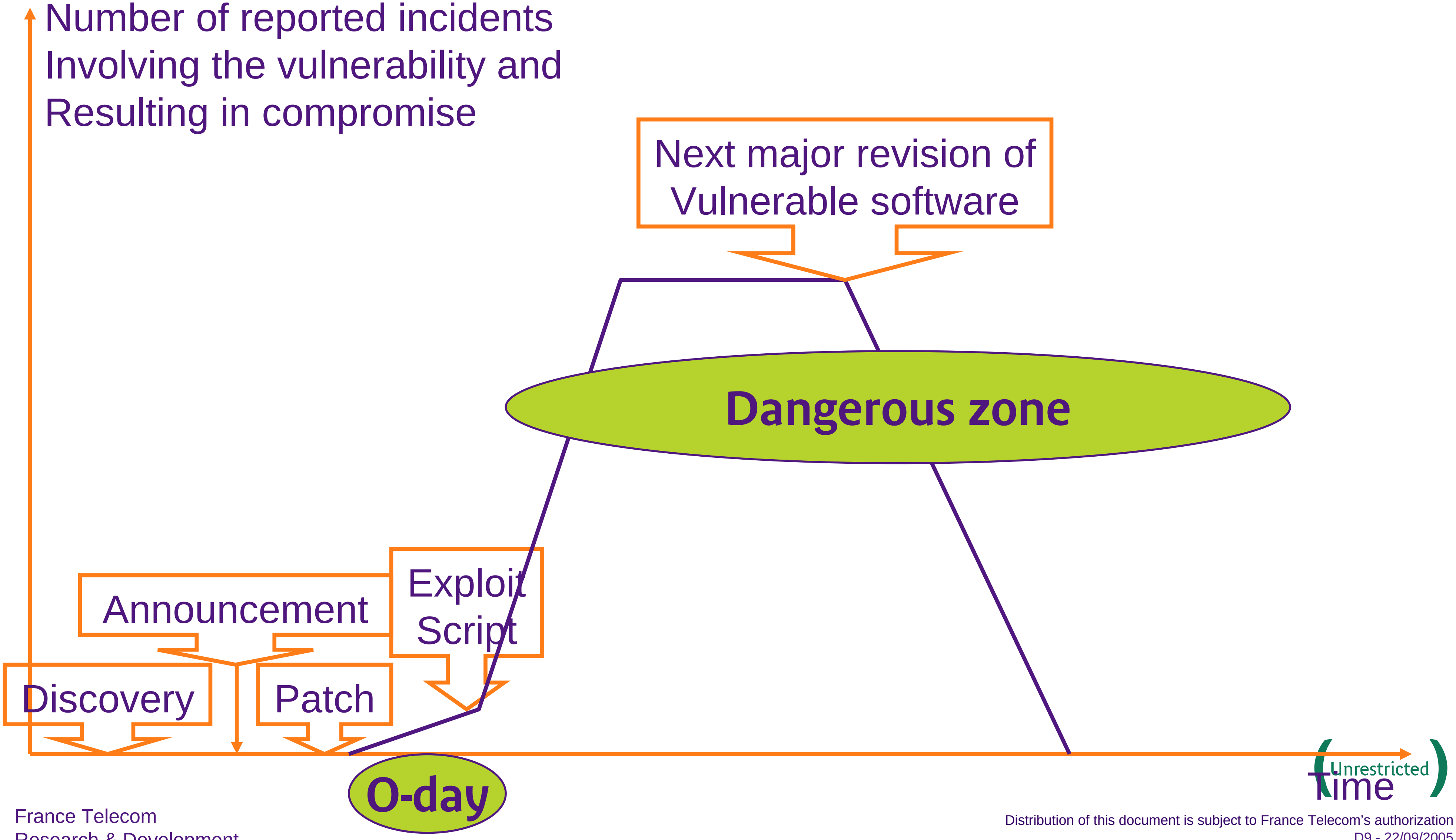
Usage errors

- Phishing
- Pharming (e.g. domain name highjacking)
- Homograph attacks (e.g. International Domain Name (IDN) issue)

System management errors

- Firewall configuration loopholes

A vulnerability life cycle



Vulnerabilities characteristics

Service or application

- Underlying operating system or distribution
- Version

Consequences

- Denial of service (crash)
- Read access to information
- Execute arbitrary code

Pre-requisites

- Local access (account, …) -> privileges escalation
- Remote access

Frequent classification:

- R2R (« Remote to Root »): administrative privileges
- R2L (« Remote to Local »): user account
- U2R (« User to Root »): privileges in local system

(Unrestricted)

Input Sanitation

- **Cross-Site Scripting: use a web server to as attack vector**
  - Enter active content in web forms
  - Input is not properly sanitized for tags
  - Content is executed on recipient's machine
  - Vulnerable sites at some point in time:
    - Webmails (yahoo, hotmail, …) on body, subject, headers
    - News and comment sites (slashdot, amazon, …)

Read hidden data
Execute code

- **Input validation errors**
  - Outlook control characters in subject lines
  - "begin" uu{de|en}code detection
  - <script> tag in message body

Execute code

- **Sometimes errors (Cert advisory CA-1997-25 Sanitizing User-Supplied Data in CGI Scripts)**
- **Still important especially when using web interfaces to legacy applications**

(Unrestricted)

Buffer Overflow

- **Mistakes in dynamic memory management**

```
foo = malloc(16*sizeof(char));  ←Allocated on the HEAP
char bar[16];                  ←Allocated on the STACK
```

- **No separation between executable and data in the stack**

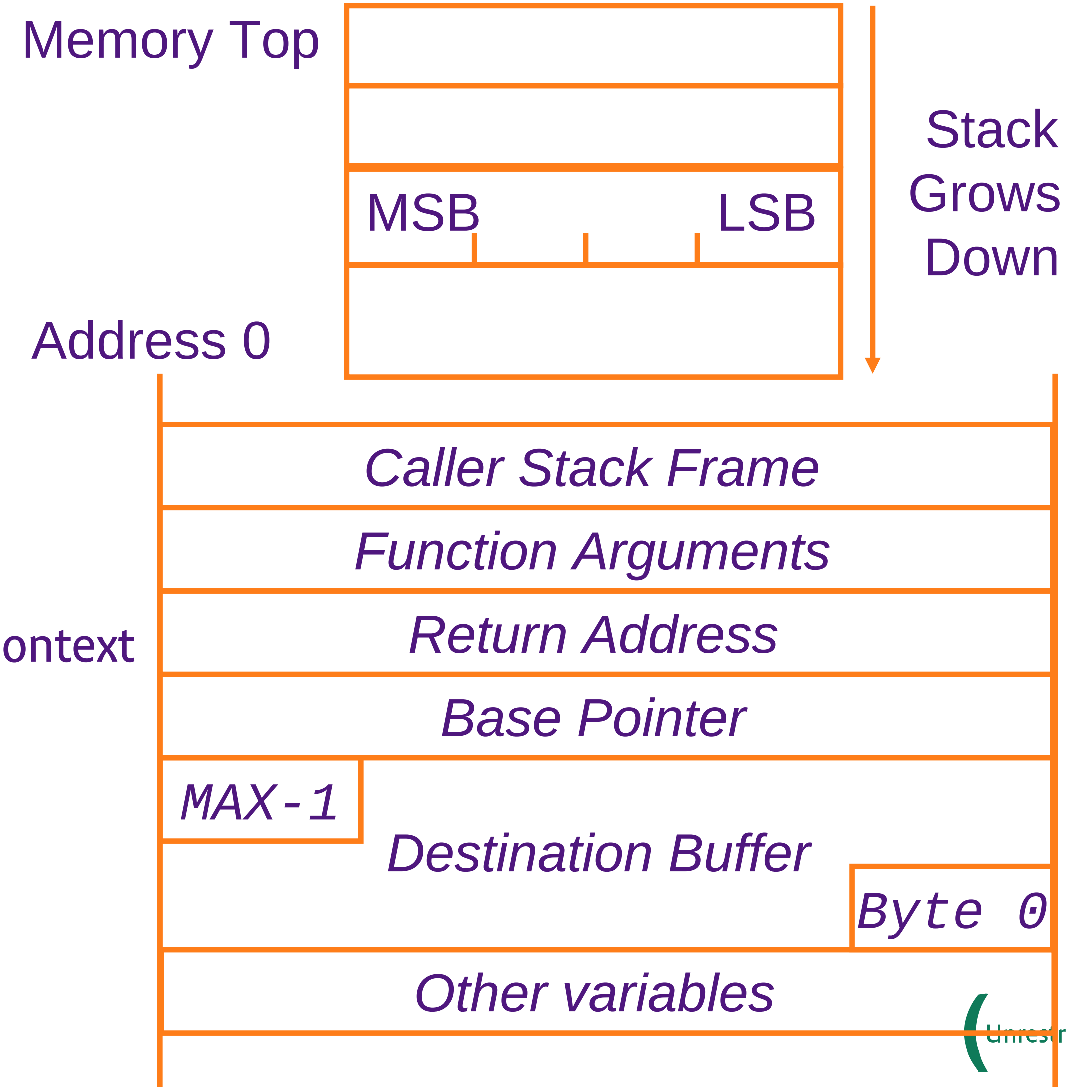
- **foo or bar contains user defined or controlled data**
  - User input
  - Environment variables
  - Remotely provided data
- **The amount of bytes read is larger than the allocated size of the buffer (read until end of string/line).**
- **Classic targets :**
  - String manipulation functions in C such as strcat, strcpy (use strncat, strncpy)
  - Input/output functions in C such as printf, scanf (use vnsprintf, …)
- **Impact: execute code with the privileges of the overflowed process**

(Unrestricted)

Shellcode Exploits: x86 Stack

- **Function call**
  - Prolog
    - Context store
    - Static data creation
  - Function code execution
  - End
    - Context restore
- **Issue**
  - Overwrite of the return context

```
Char foo(char arg1) {
      char foo[MAX];
      …
      return(‘a’);
}
```



Memory Top

MSB LSB

Address 0

Stack Grows Down

Caller Stack Frame

Function Arguments

Return Address

Base Pointer

MAX-1

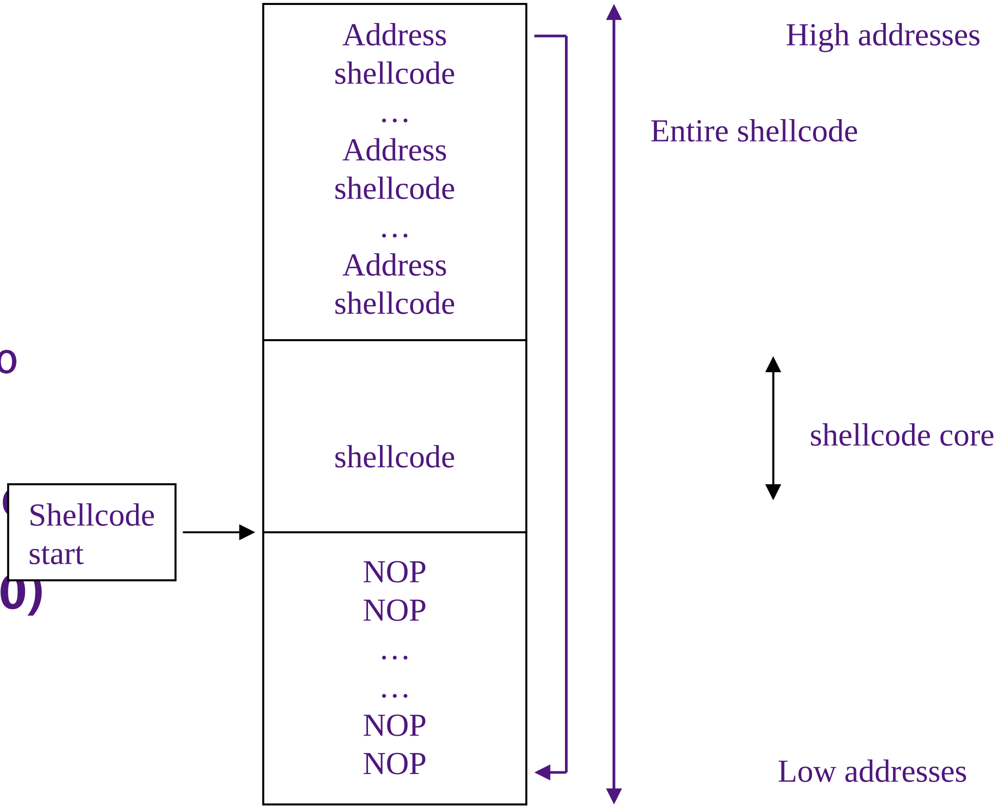
Destination Buffer

Byte 0

Other variables

Shellcode Exploits: Layout

- **3 parts**
  - NOPs
    - Space eater
  - Shellcode body
    - The malicious code
  - Address pointer
    - Sends execution pointer to shellcode body
- **Cannot contain end-of-line**
- **Cannot contain « null » (\0)**



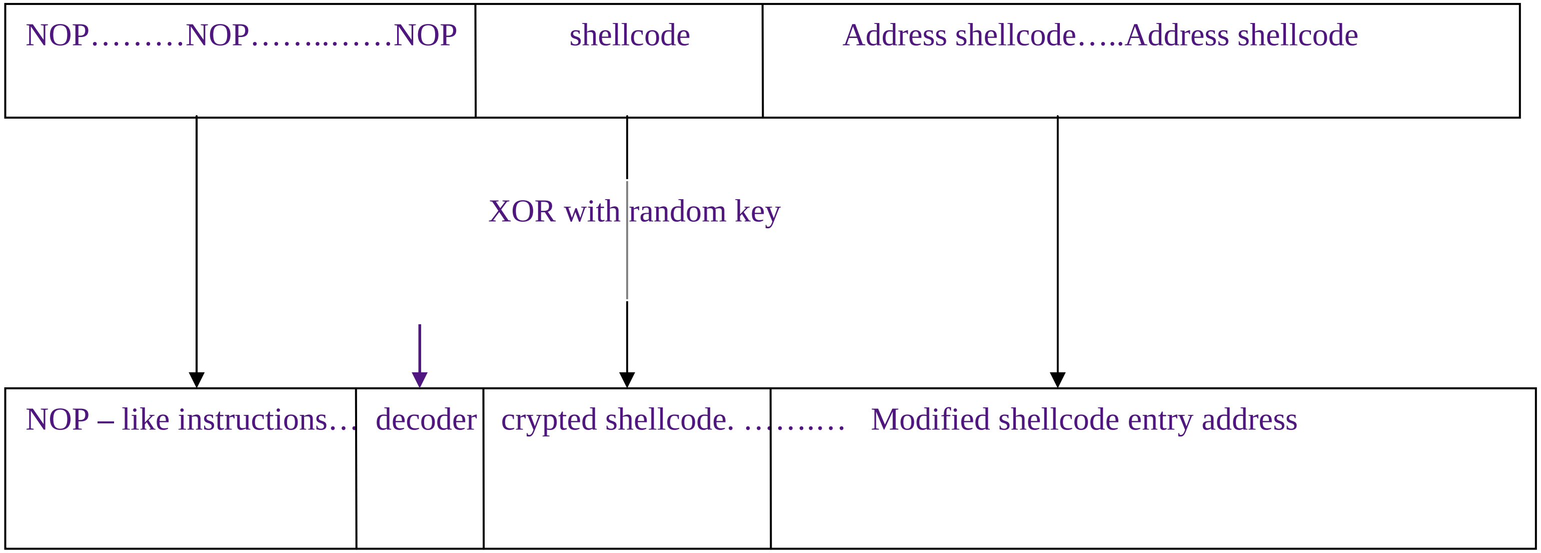
(Unrestricted)

Shellcode Exploits: “ADMutate”

- **« Polymorphic » shellcode**
  - Similar in principle to polymorphic viruses
- **The executed shellcode looks different for each attack instance.**
- **4 elements in the shellcode instead of 3.**
- **Use of « equivalent instruction set » libraries.**
- **Likely to be limited to CISC (Pentium-like) processors**
  - RISC processors have limited instruction sets, hence limited libraries.
  - However, ADMutate also works for SPARC (SUN).
- **Requires a precise understanding of the instruction set of each processor.**
- **Really easy to use**
  - Insert a C API inside the attack code that creates the shellcode.
  - Define specific structures
  - Call three functions to create the polymorphic shellcode.

(Unrestricted)

Polymorphic Shellcode



NOP………NOP……..……NOP | shellcode | Address shellcode…..Address shellcode

XOR with random key

NOP – like instructions… | decoder | crypted shellcode. ……..… | Modified shellcode entry address

Format String Vulnerabilities

- **Wide discovery in 1999-2000**
- **Mistake**
  - Normal syntax: printf("%s", $var);
  - Replaced with: printf($var);
- **Problem: if $var is user-defined**
  - Input data, command line
  - Environment variable
  - Syslog message
- **Impact: run code with the privileges of the vulnerable process**
- **All functions using format strings are vulnerable :**
  - ({v,f,vf,sn}{scanf,printf}, syslog, …).
- **The trick : Use %n to write chosen data on the stack**
- **The amount of freedom given to the attacker is less than for buffer overflows**
- **It is still possible to carry out « remote to root » attacks**

(Unrestricted)

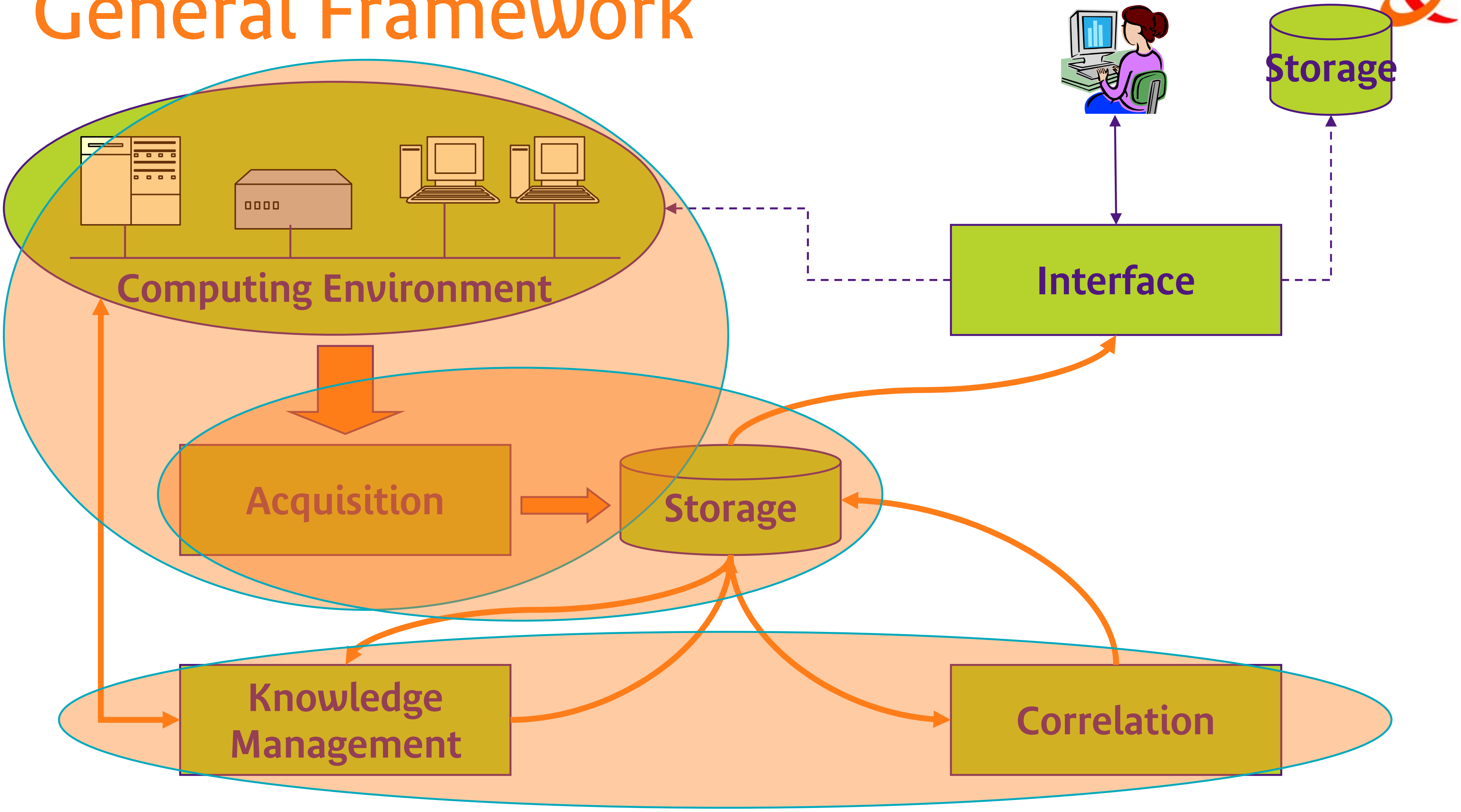
Race Condition Vulnerabilities

- **Abuse the scheduling mechanism of modern operating systems**
- **Transactions span multiple system calls and hence are not atomic**

```
if (stat ($file, & st) < 0)
 {fprintf (stderr, "%s not found\n", $file); exit(EXIT_FAILURE); }
if (st . st_uid != getuid ())
 {fprintf (stderr, "%s does not belong to you !\n", $file);
 exit(EXIT_FAILURE); }
if (! S_ISREG (st.st_mode))
 {fprintf (stderr, "%s is not a normal file\n", $file); exit(EXIT_FAILURE); }
if ((fp = fopen ($file, "w")) == NULL)
 {fprintf (stderr, "Impossible to open %s\n", $file); exit(EXIT_FAILURE);}
```

- **The fault occurs when an attacker changes the inode/filename relationship**
  - Symlink to some other file
- **Exploits look and are difficult and random**
  - Some of them require a number of tries to succeed.
  - Techniques can be used to improve the success rate (overloading, …)

General Framework



Unrestricted

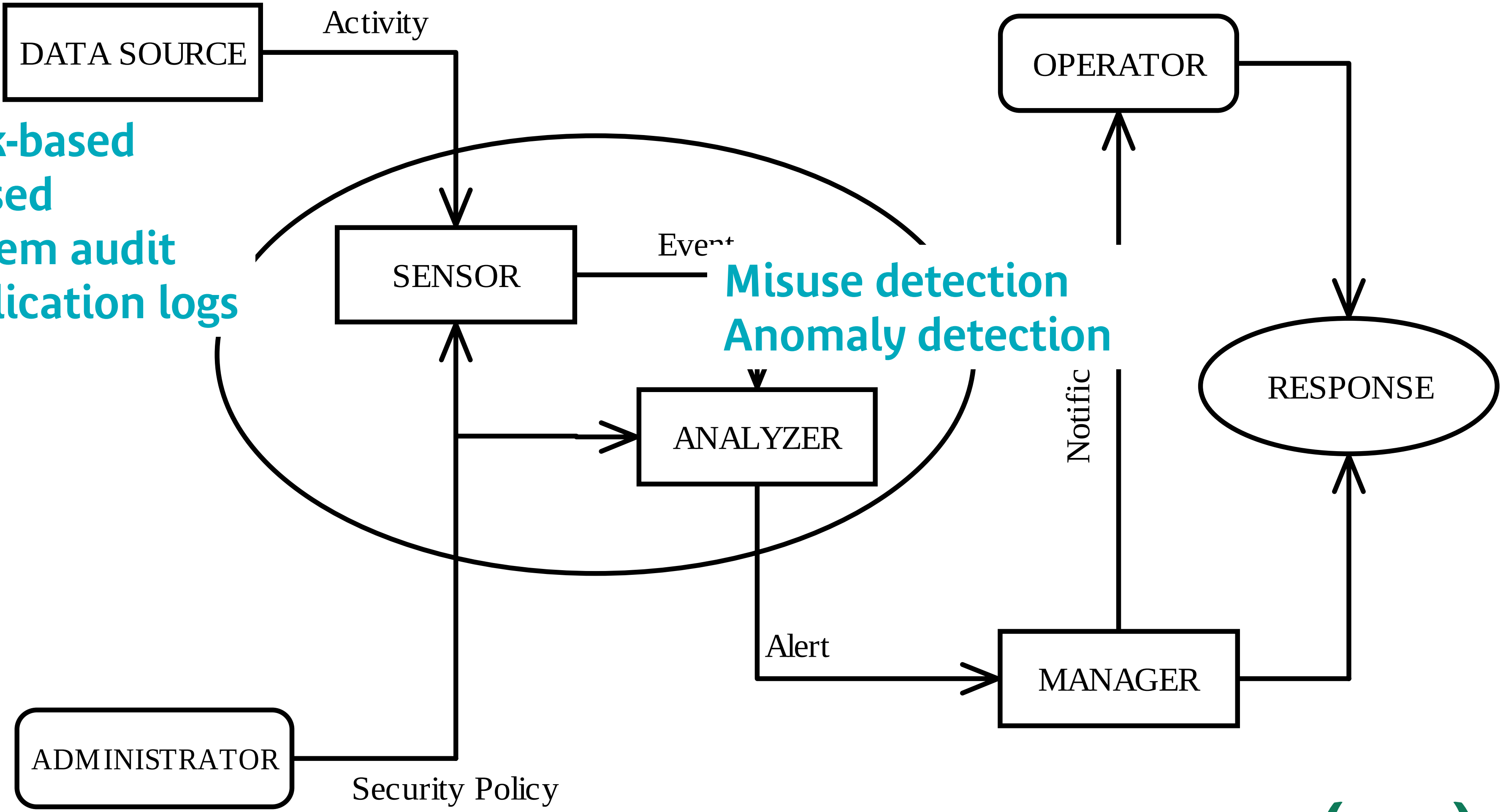
Distribution of this document is subject to France Telecom's authorization

Intrusion detection

- **Analyze activity occurring on an information system**
- **Captured through traces**
  - Network packets
  - System audit information
  - Application logs
- **To detect malicious activity**
  - Misuse detection
  - Anomaly detection
- **IDS sensors**
  - Complement data acquisition processes (existing logs)
  - Replace obscure logs (e.g. fw logs)

(Unrestricted)

Standard Architecture (IDWG)



Network-based

Host-based

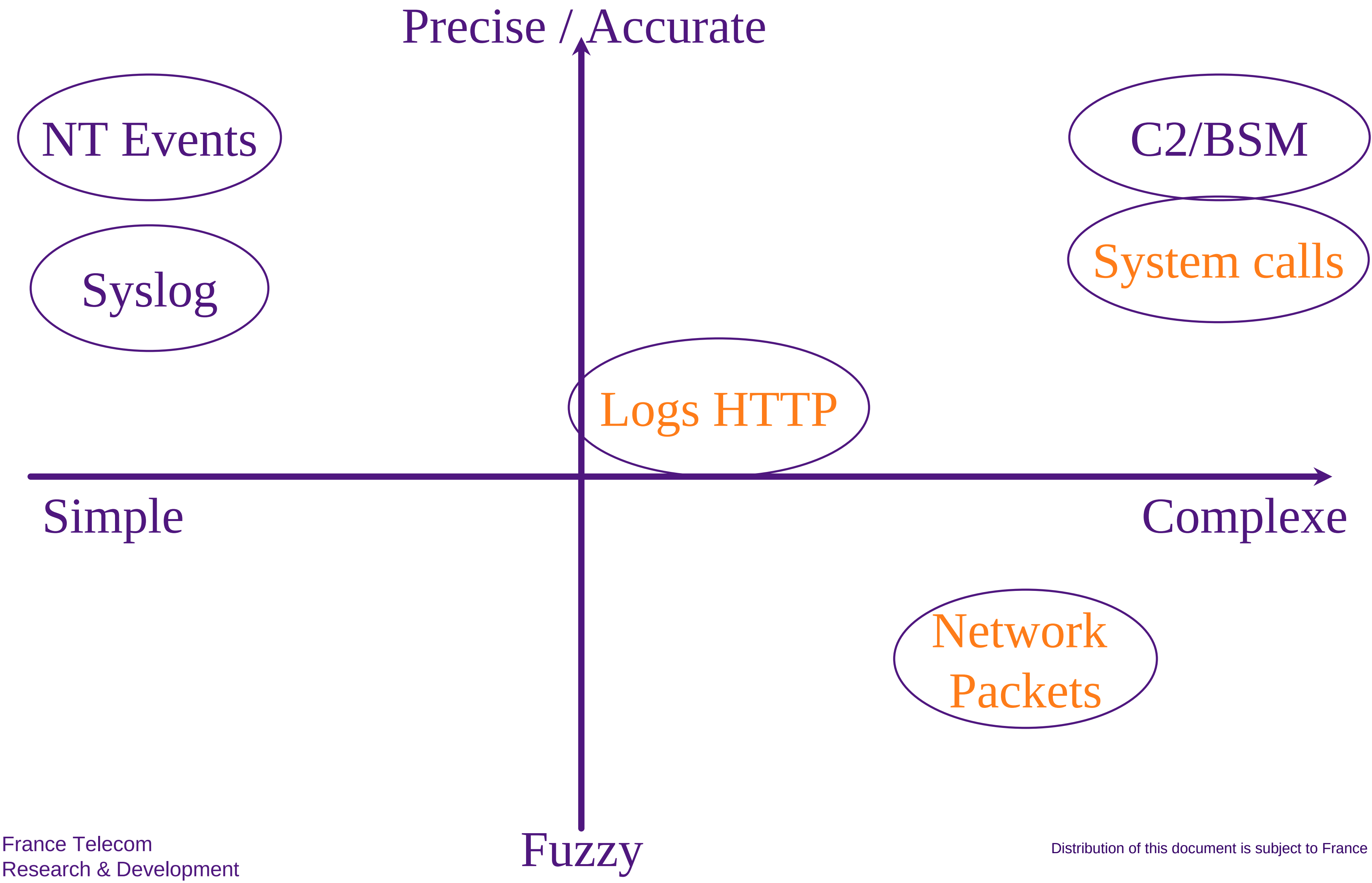
- System audit
- Application logs

Misuse detection

Anomaly detection

Unrestricted

Complexity of the Data Sources



(Unrestricted)

Anomaly detection

- **Model the expected behaviour of the information system**
  - Explicit model (e.g. given by a security policy)
  - Model acquired through learning algorithms
- **Alert raised when current behaviour does not match**

- **Advantages**
  - Possibility to detect usage of previously unknown vulnerabilities (wider monitoring, masquerading).
- **Drawbacks**
  - Difficulty to train or define models
  - Difficult to validate models for completeness
  - Difficulty to understand alerts and propose countermeasures

(Unrestricted)

Example: STIDE

- Sequence TIme Delay Embedding [Forrest96]
  - Model the execution of a process via sequences of system calls
    - Or library calls, jumps, …
  - New sequences represent anomalies and generate alerts
- Multiple approaches for generating model
  - Observation of executions
  - Decompilation of code
- Multiple approaches for describing model
  - Strings of symbols
  - Finite state automata
- ⊕ Many theoretical studies show interesting detection rates
- ⊖ Important overhead makes real usage difficult

STIDE: example and limitations

Simple copy program

```
main() {
    fstat(src);
    fstat(dst);
    open(src);
    open(dst);
    while(read(buf,src)) {
        write(buf,dst);
    }
    close(src);
    close(dst);
}
```

Main, fstat, fstat, open, open, read, write, read, close, close

main, fstat

main, fstat, fstat, open, open, read, close, close

main, fstat, fstat, open, open, open, read, write, read, write, read, close, close

main, fstat
fstat, fstat
fstat, open
open, open
open,read
read, write
write, read
read, close
close, close

Magic number : 6

Variable length patterns

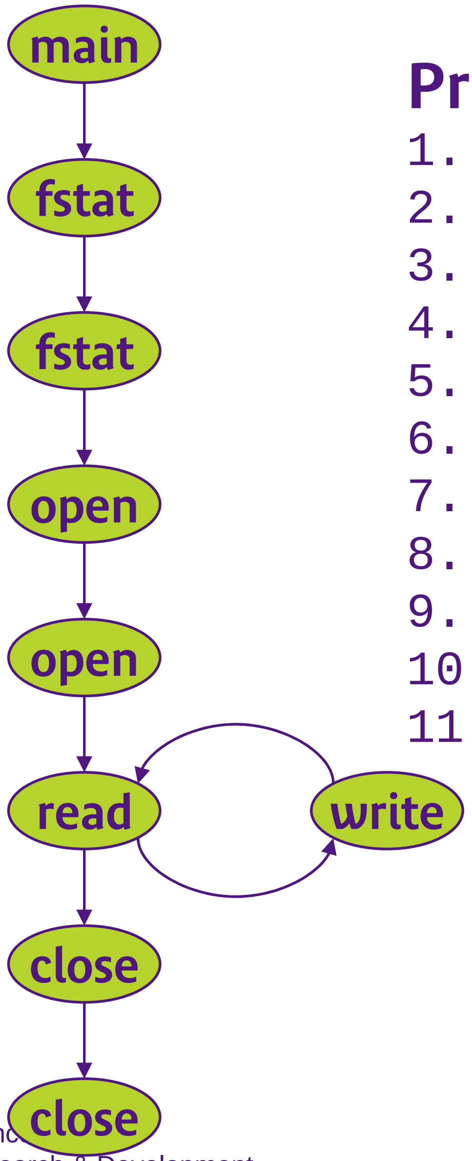
Multiple representations

- Strings of system calls
- Finite State Automata (NFSA, DFSA)

Still ongoing research

(Unrestricted)

Current research and issues



Problem 1

```
if (c) {
    a;
} else {
    b;
}
// Some process
if (c) {
    c;
} else {
    d;
}
```

FSA may contain ac and bd, but also ad and bc.

Problem 2

```
// function g()
S0;
If(!superuser){
    f();
    return;
}
f();
S1;
//function f()
overflows;
S2;
```

Problem 3

```
//function g()
If (!superuser) {
    f();
    return;
}
// become superuser
execve("/bin/sh");
//function f()
no syscall but overflows;
```

Unrestricted

Distribution of this document is subject to France Telecom's authorization

Misuse detection

- **Search for evidence of known malicious activity**
  - String of events
  - Pattern matching
- **Alert generation condition is explicit**

- **Advantages**
  - Alerts easy to document
  - Countermeasures proposed
- **Drawbacks**
  - Difficult to detect 0-day, new exploits
  - Difficult to follow the evolution of knowledge
  - Difficult to detect configuration errors

(Unrestricted)

PHF vulnerability

- phf Remote Command Execution Vulnerability
- http://www.securityfocus.com/bid/629
- bugtraq id 629
- CVE-1999-0067
- Vulnerable
  - Apache Software Foundation Apache 1.0.3
  - NCSA httpd 1.5 a-export
- Not vulnerable
  - ?
- Exploit : run commands on system as uid of the server
- How to detect this from the server ? From the network ?

(Unrestricted)

Analysis of a web server log

- **3 step process**
  - Normalization: read line, decode, segment
  - Feature extraction: regular expressions
  - Reconciliation: prolog rules
- **Objective: accurate diagnostic**
  - Signatures organized in classes
- **Requests (http://www.securityfocus.com/bid/629):**
  - `/cgi-bin/phf`
  - `/cgi-bin/phf?/etc/passwd`
  - `/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd`
- **Status codes**
  - 404: not found
  - 403: authentication requested
  - 200: success

(Unrestricted)

Example (Pattern rule)

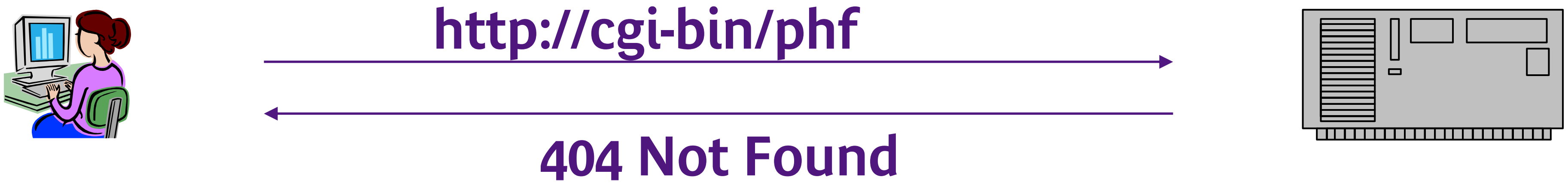
```
<signature name="phf"
            trigger="/phf$"
            severity="+1"
            class="query,apache,cgi" >
 <description origin="cve">
   <name>CVE-1999-0067</name>
   <url>http://cve.mitre.org/cgi-
  bin/cvename.cgi?name=CVE-1999-0067</url>
   </description>
   <description origin="bugtraqid">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

Example (Prolog rules)

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

<signature name="failed_cgi"
           trigger="pattern(notfound_404),class(cgi),
                  !pattern(args_not_empty)"
           severity="-1"
           class="rule">
  <description origin="vendor-specific">
    <name>If a CGI script referenced as dangerous has an explicit
    failed status, then the severity is decreased. </name>
    <url>file://./signatures.xml</url>
  </description>
</signature>
```

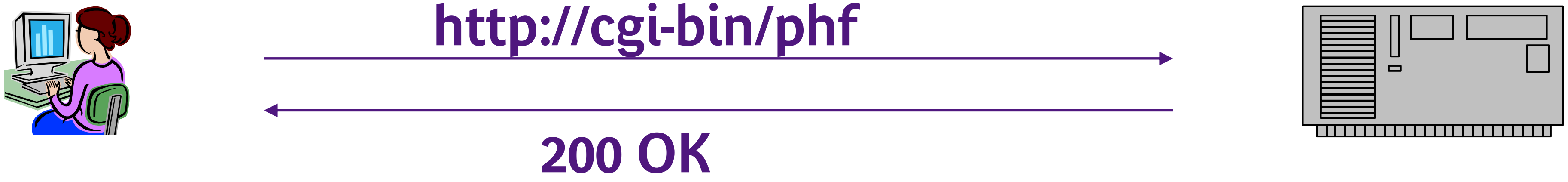
Example (1)



```
1.1.1.1 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf HTTP/1.0" 404 310
```

Severity :	0	
notfound_404	0	
get_method	0	
cgi_dir	0	
phf (cgi_dir)	1	(implies cgi)
failed_cgi	-1	(cgi + notfound_404)

Example (2)

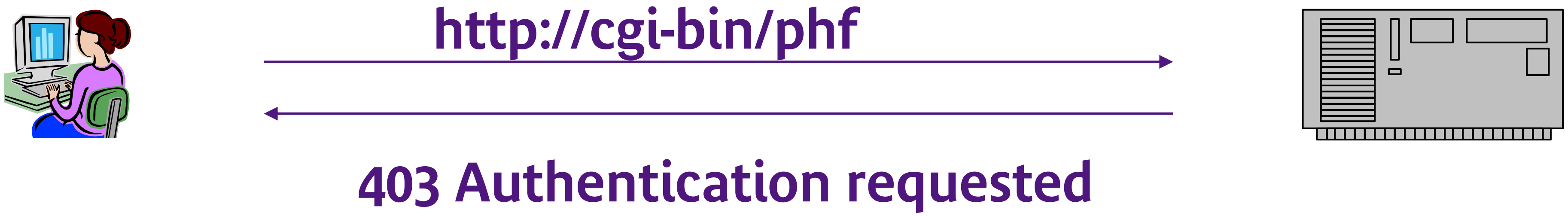

http://cgi-bin/phf

200 OK

```
1.1.1.2 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf HTTP/1.0" 200 310
```

Severity :	4	
status_200	0	
get_method	0	
cgi_dir	0	
phf (cgi_dir)	+1	(implies cgi)
success_cgi	+3	(cgi + status_200)

Example (3)



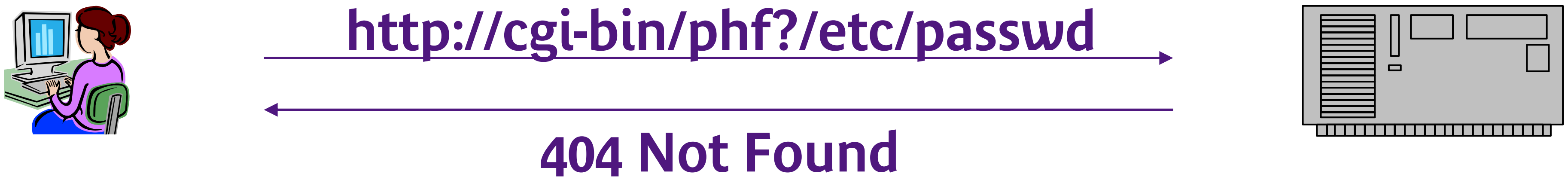
http://cgi-bin/phf

403 Authentication requested

```
1.1.1.3 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf HTTP/1.0" 403 129
```

Severity :	2	
not_allowed_40x	1	
get_method	0	
cgi_dir	0	
phf (cgi_dir)	1	(implies cgi)

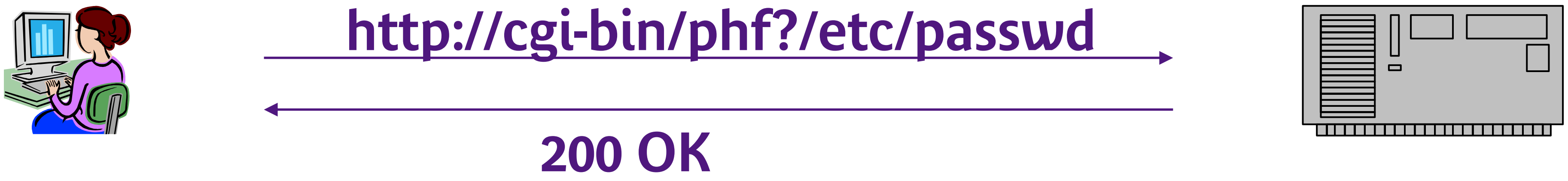
Example (4)



```
1.1.1.4 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf?/etc/passwd HTTP/1.0" 404 310
```

Severity :	2	
notfound_404	0	
get_method	0	
cgi_dir	0	
phf (cgi_dir)	1	(implies cgi)
etc_password	1	(implies file)
args_not_empty	0	
real_attempt	2	(cgi + file)
failed_cgi	-1	(cgi + notfound_404)
failed_file	-1	(file + notfound_404)

Example (5)

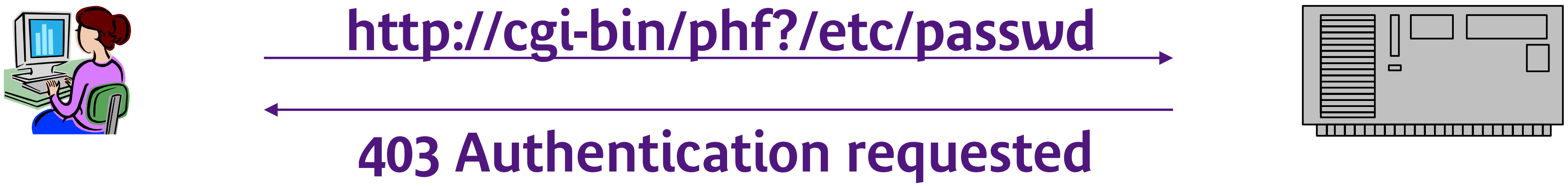


```
1.1.1.5 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf?/etc/passwd HTTP/1.0" 200 2450
```

Severity :	10	
status_200	0	
get_method	0	
cgi_dir	0	
phf (cgi_dir)	1	(implies cgi)
etc_password	1	(implies file)
args_not_empty	0	
real_attempt	2	(cgi + file)
success_cgi	+3	(cgi + status_200)
success_file	+3	(file + status 200)

(Unrestricted)

Example (6)



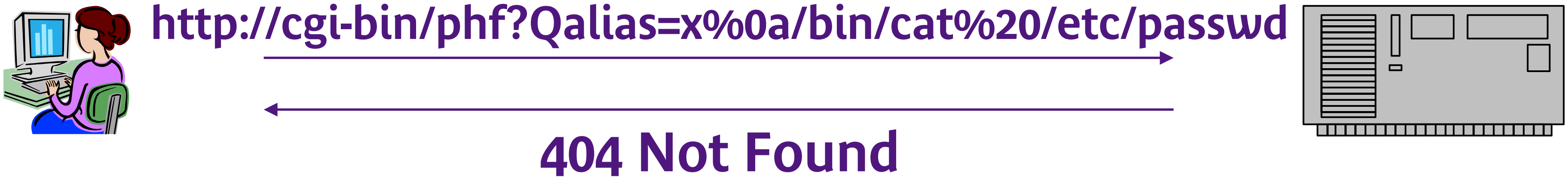
http://cgi-bin/phf?/etc/passwd

403 Authentication requested

```
1.1.1.6 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf?/etc/passwd HTTP/1.0" 403 129
```

Severity :	**5**	
not_allowed_40x	1	
get_method	0	
cgi_dir	0	
phf (cgi_dir)	1	(implies cgi)
etc_password	1	(implies file)
args_not_empty	0	
real_attempt	2	(cgi + file)

Example (7)

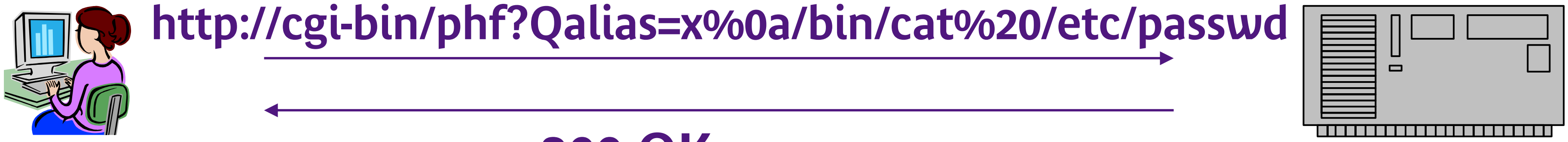

http://cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd

404 Not Found

```
.1.1.7 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0" 404 3
```

Severity :	4	
non_ascii	1	
notfound_404	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)
failed_cgi	-1	(cgi + notfound_404)
failed_file	-1	(file + notfound_404)

Example (8) (successful attack …)



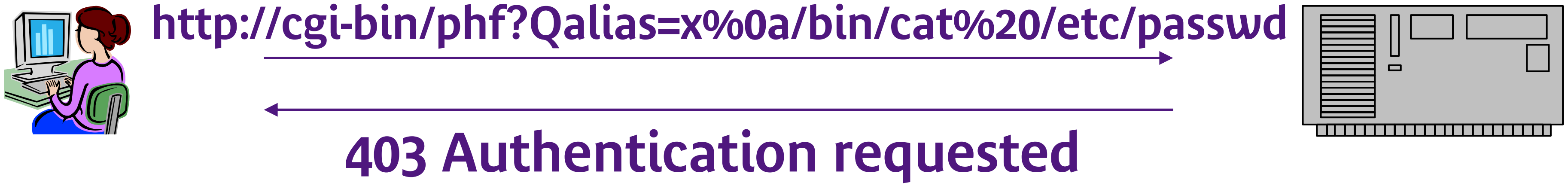
http://cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd

200 OK

```
.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 2
```

Severity :	12 (from 10)	
non_ascii	1	
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)

Example (9)



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

403 Authentication requested

```
.1.1.9 - - [26/Feb/2002:18:37:19 -0500] "GET /cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd HTTP/1.0" 403 3
```

Severity :	7	
non_ascii	1	
not_allowed_40x	1	
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)

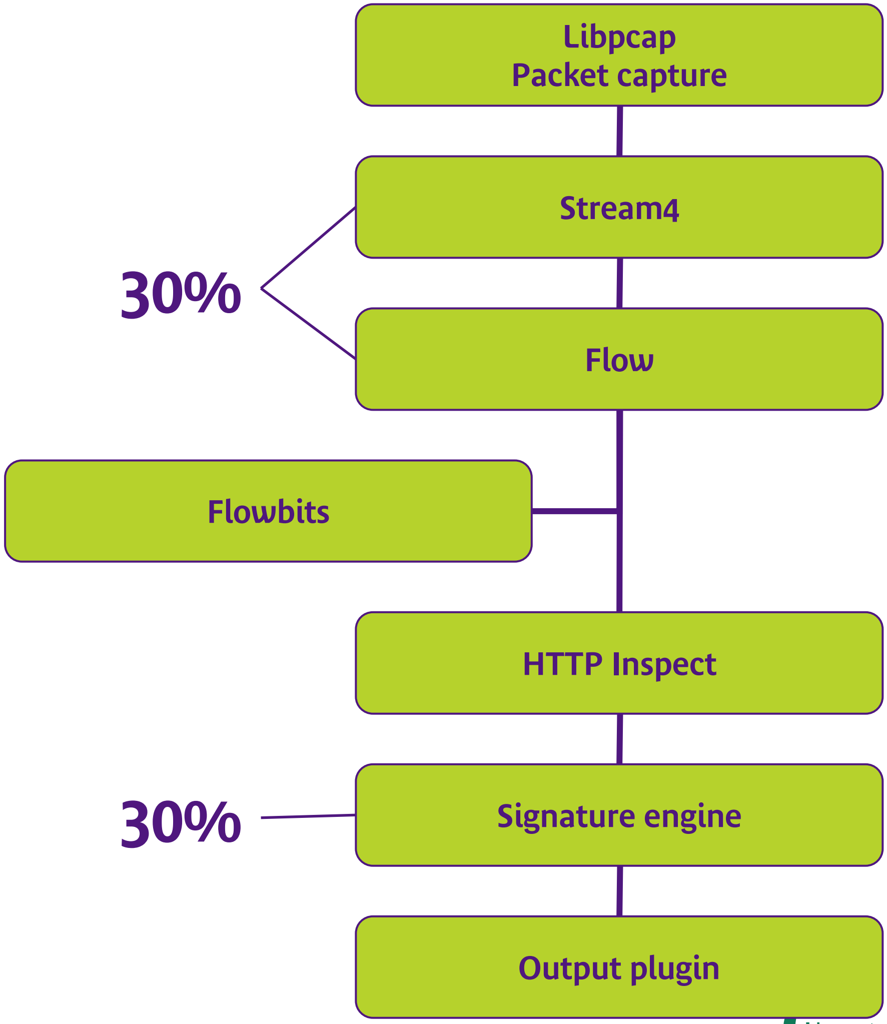
Overview of WebAnalyser

- **664 signatures that recognize**
  - Attacks (~50%)
  - Attack hints (e.g. evasive actions, perl code, …)
  - Attack contexts (e.g. method, status code)
- **Diagnosis based on continuous severity value**
- **4 classes of output:**
  - C0: S=0, normal
  - C1: S in [1,4], abnormal encodings and unsuccessful attacks
  - C2: in between, possibly successful, no automated interpretation possible
  - C3: S in [9, …], definitively successful attacks

(Unrestricted)

Snort (http://www.snort.org)

- Network based IDS
- Misuse IDS
- Pre-processors
  - Rebuild the event stream
  - IP fragmentation
  - TCP, UDP fragmentation
  - Counters
- Signature engine
  - Packet headers (RTN)
  - Packet flags + content (OTN)
  - Context
- Multiple outputs



Unrestricted

Distribution of this document is subject to France Telecom's authorization

Snort signatures format

- **Snort signatures are becoming a de-facto standard for**
  - The IDS industry
  - The anti-virus industry
- **Fields:**
  - Type of signature = alert|pass|log|dynamic (deprecated by tags)
  - Protocol = any|tcp|udp|icmp|…
  - Source and destination adresses and ports
  - Options
    - What should be found (or not found) in the data and where (limited)
    - What to send to the manager (message, references, identifiers)
    - Memorize and retrieve states (flows, flowbits)

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Miscellaneous long HTTP
      WebDAV request"; content:" /"; content:!"|0a|"; within:60000;
            flow:to_server; reference:Bugtraq,7116; rev: 2; )
```

(Unrestricted)

Equivalent Snort rules

Network Intrusion Detection

- Need to process multiple packets

Snort detection process

- Multiple pre-processors
  - Stream4
  - Flows
  - http inspect
- Rule engine

Could match /phfqalias
Does not know the unix command
Short-circuits the passwd rule

Snort PHF rules

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI
phf arbitrary command execution attempt";flow:to_server,established;
uricontent:"/phf"; nocase; content:"QALIAS"; nocase; content:"%0a/";
reference:bugtraq,629; reference:arachnids,128; reference:cve,CVE-1999-
0067; classtype:web-application-attack; sid:1762; rev:1;)
```

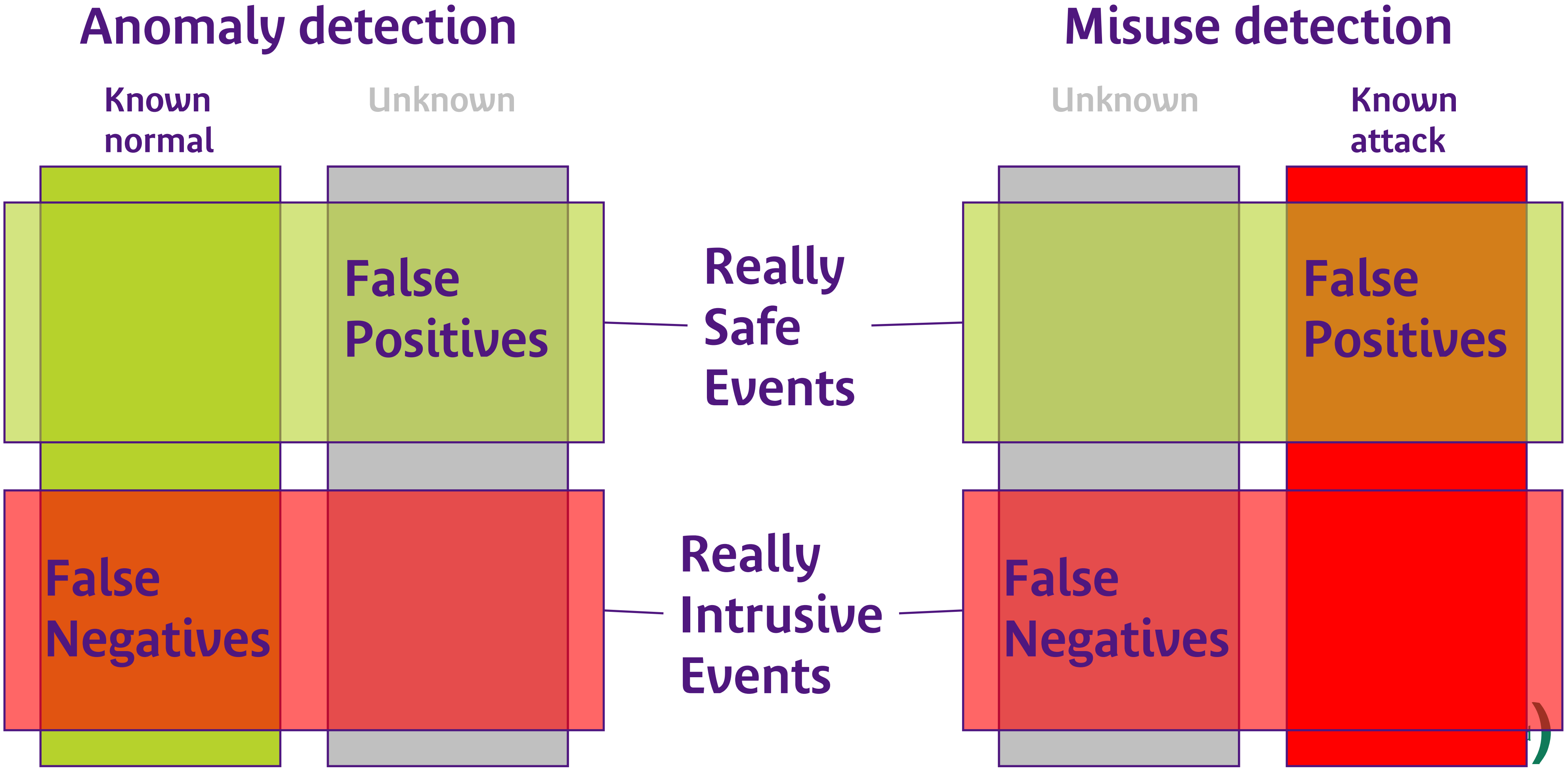
```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-CGI
phf access";flow:to_server,established; uricontent:"/phf"; nocase;
reference:bugtraq,629; reference:arachnids,128; reference:cve,CVE-1999-
0067;  classtype:web-application-activity; sid:886; rev:8;)
```

Snort rules assessment

- **Complex process (pre-processors)**
  - Evasion
  - Short-circuit rules
- **Separation between attempt (attack) and access (scan)**
  - Knowledge in the message
  - Not a systematic endeavour
- **Does not capture the server response**
  - Using tags from the flow pre-processor
  - Memory management issues
  - Multiply the number of rules by 3 or 4 ?
- **Good knowledge of the HTTP protocol, but others ?**
- **Separate inbound, internal and outbound activities ?**
- **Is the diagnosis really satisfactory ?**

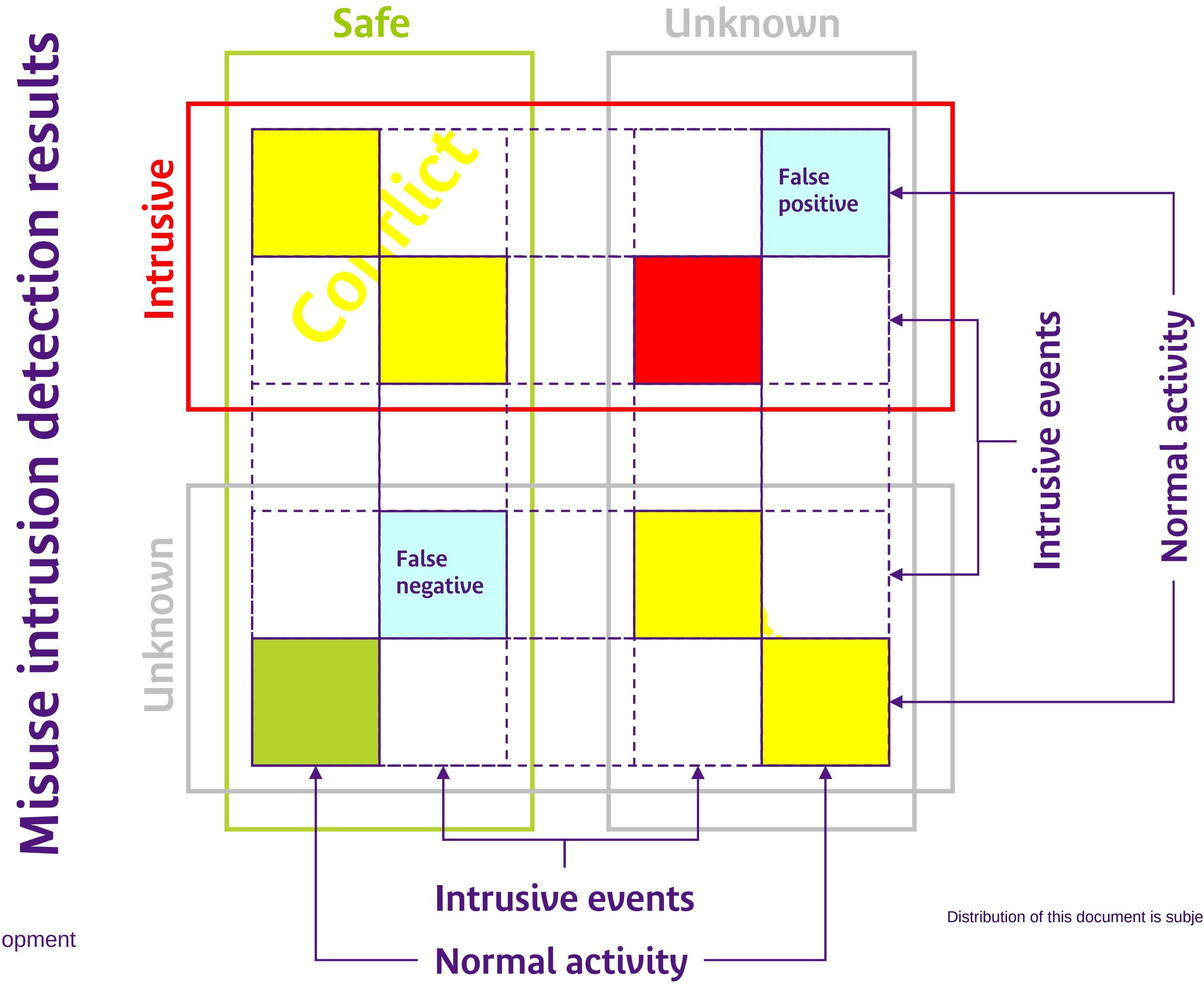
(Unrestricted)

Back to basic definitions



Flat combination (NIDES88-92)

Anomaly intrusion detection results



Safe

Unknown

Misuse intrusion detection results

Intrusive

Unknown

Conflict

False positive

False negative

Intrusive events

Normal activity

Intrusive events

Normal activity

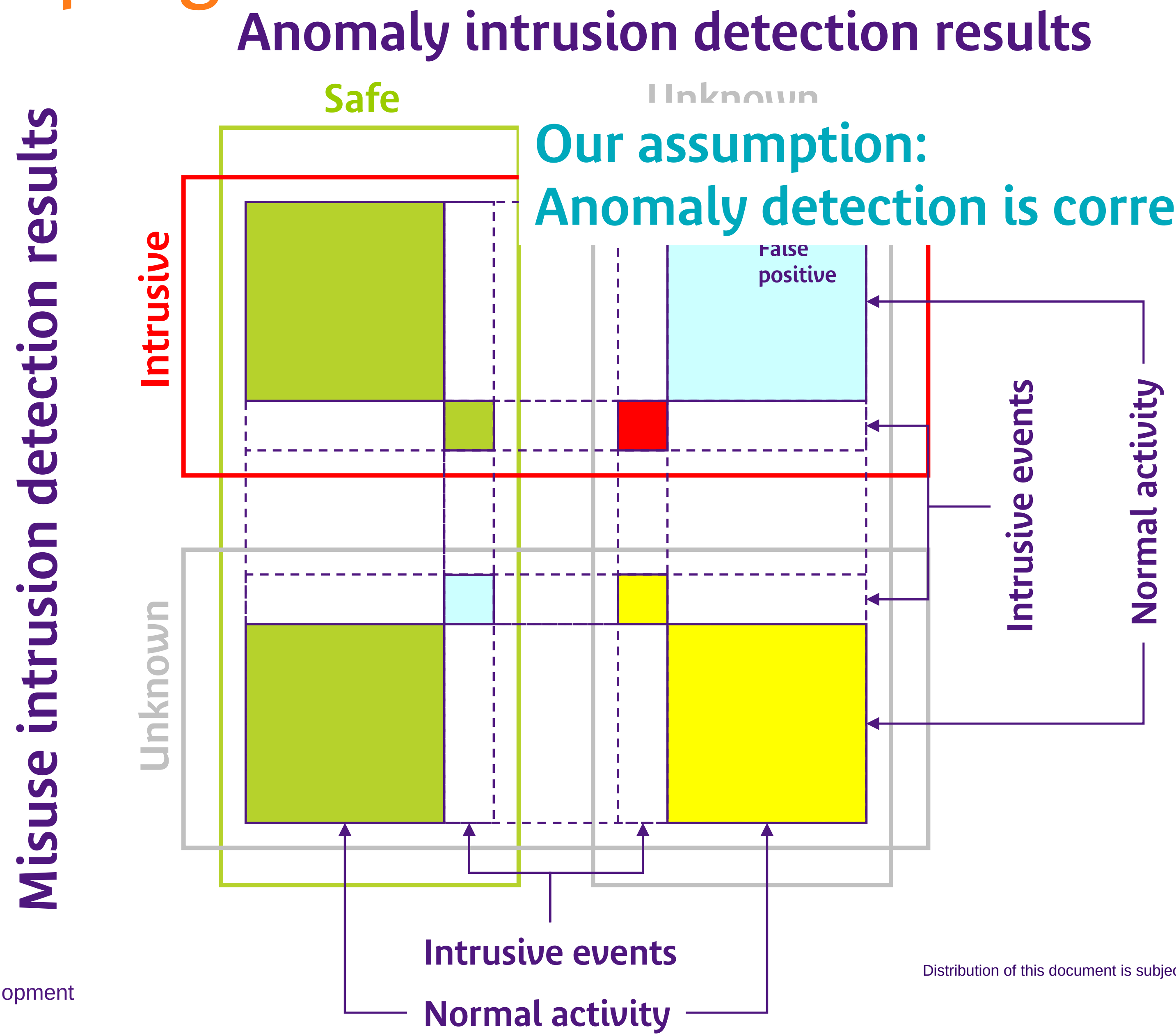
Distribution of web server logs

Diagnosis	Supélec 2003	France Télécom 2001
Normal traffic	79.14%	89,13%
Abnormal artifacts and unsuccessful attacks	20,82%	10,87%
Definite attempts, Mostly unsuccessful	0,03	0
Possibly successful attacks	0,01 %	0

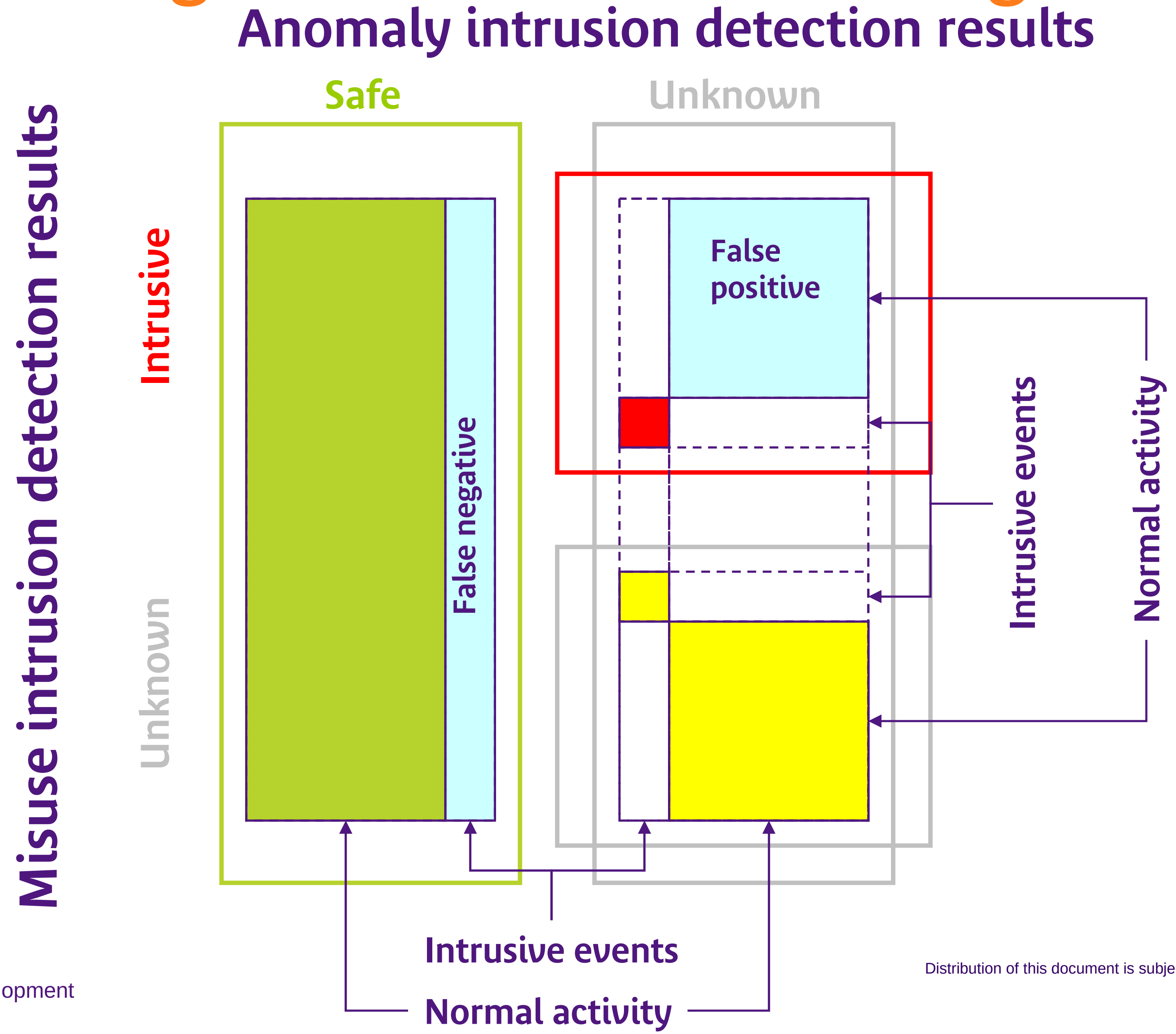
(Unrestricted)

Reshaping volumes

Our assumption:
Anomaly detection is correct on safe



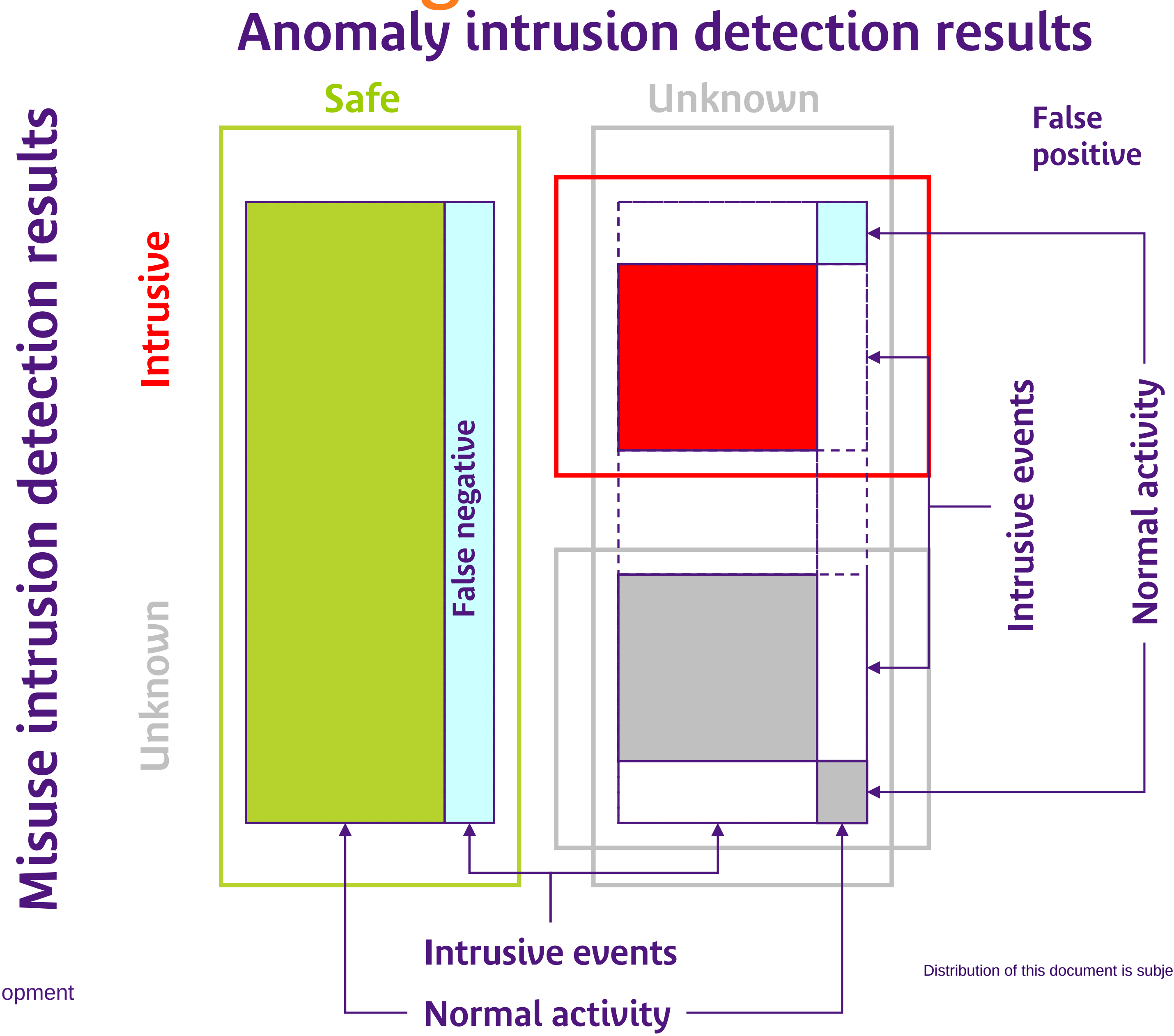
Cascading instead of combining



(Unrestricted)

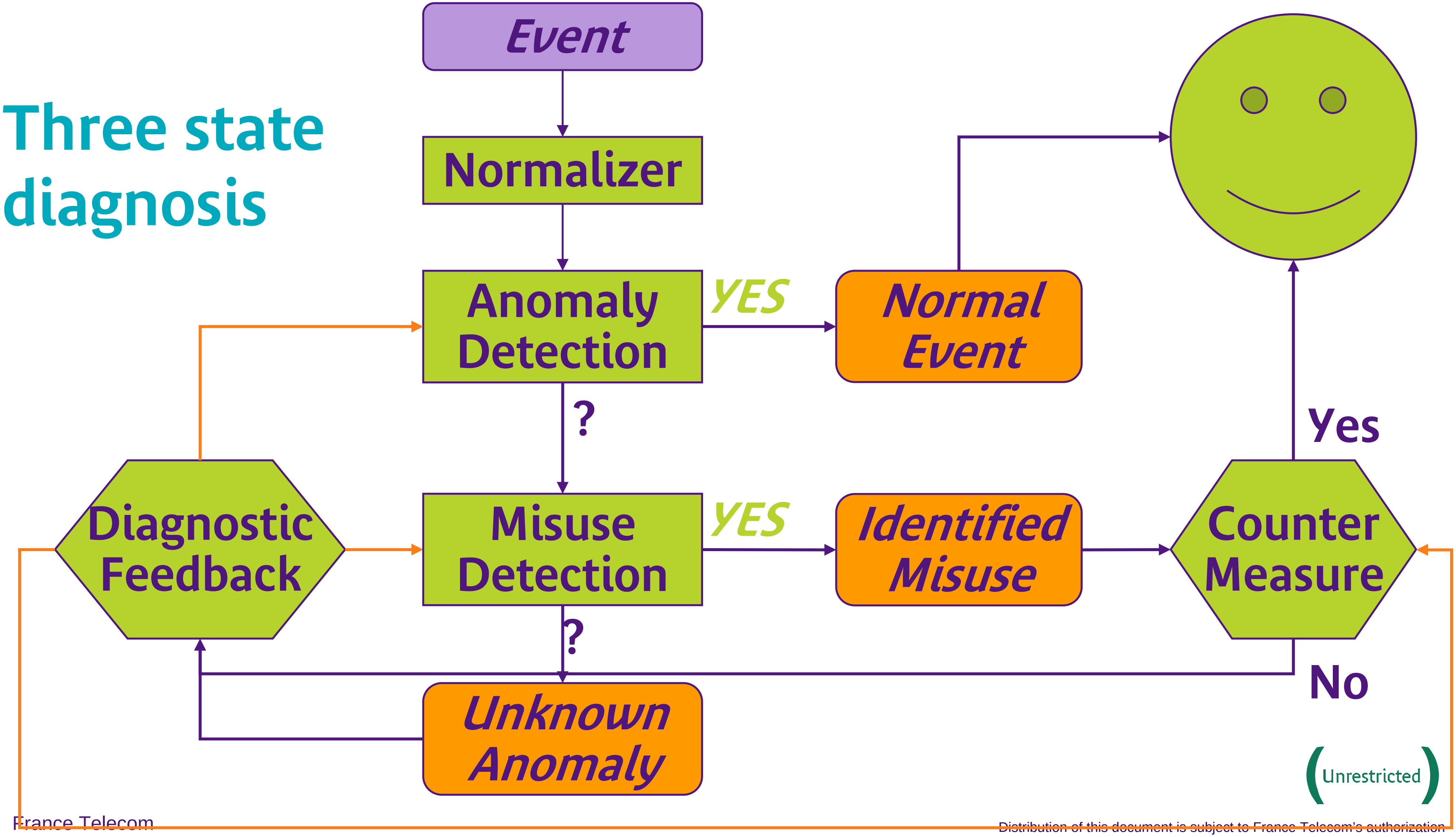
France Telecom
Research & Development

Resize and recognize unknown



Cascade architecture

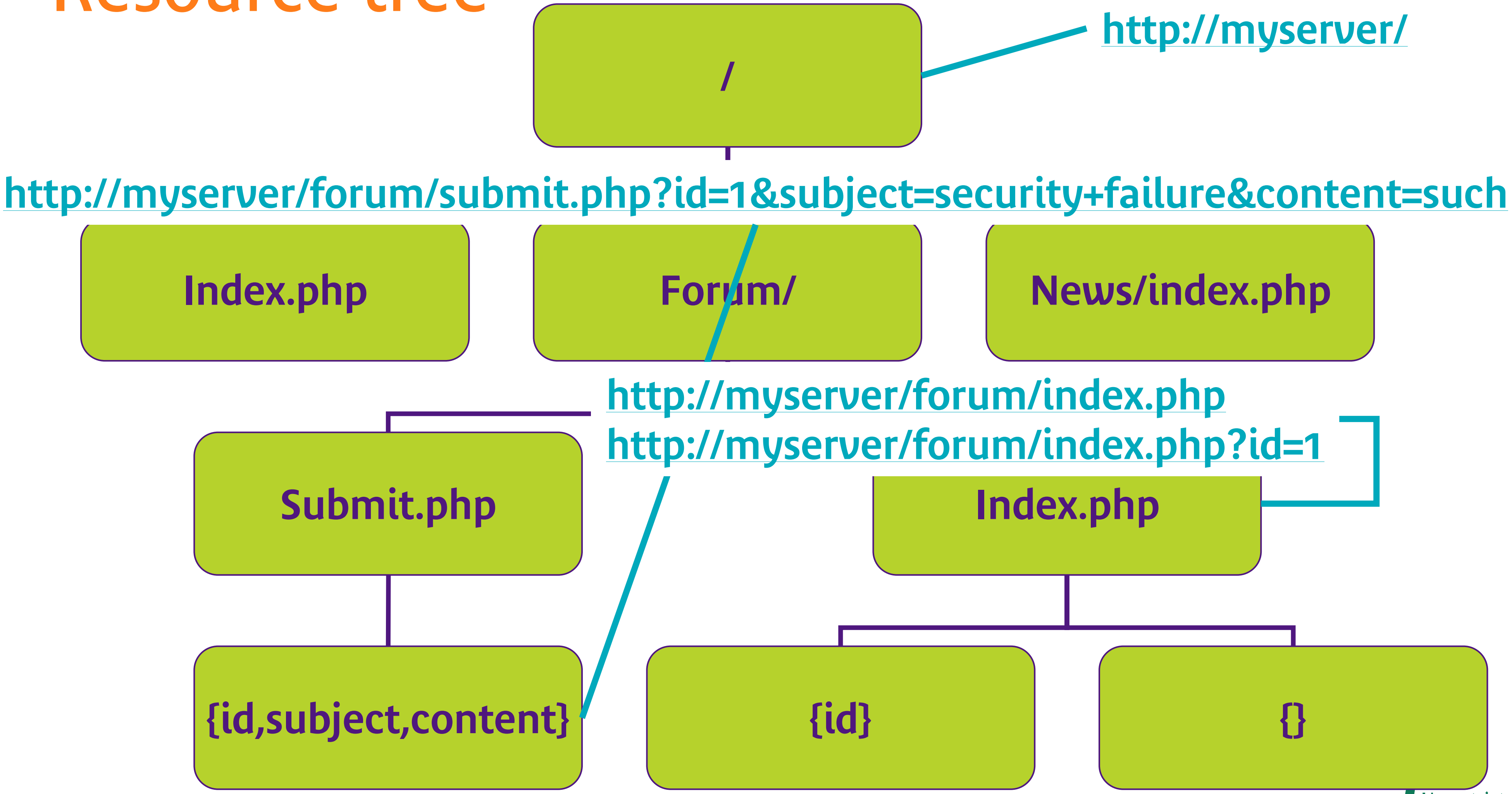
Three state diagnosis



(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Simple anomaly detection system
Resource tree


/
http://myserver/
http://myserver/forum/submit.php?id=1&subject=security+failure&content=such
Index.php
Forum/
News/index.php
http://myserver/forum/index.php
http://myserver/forum/index.php?id=1
Submit.php
Index.php
{id,subject,content}
{id}
{}

Unrestricted

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

(Unrestricted)

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

(Unrestricted)

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

(Unrestricted)

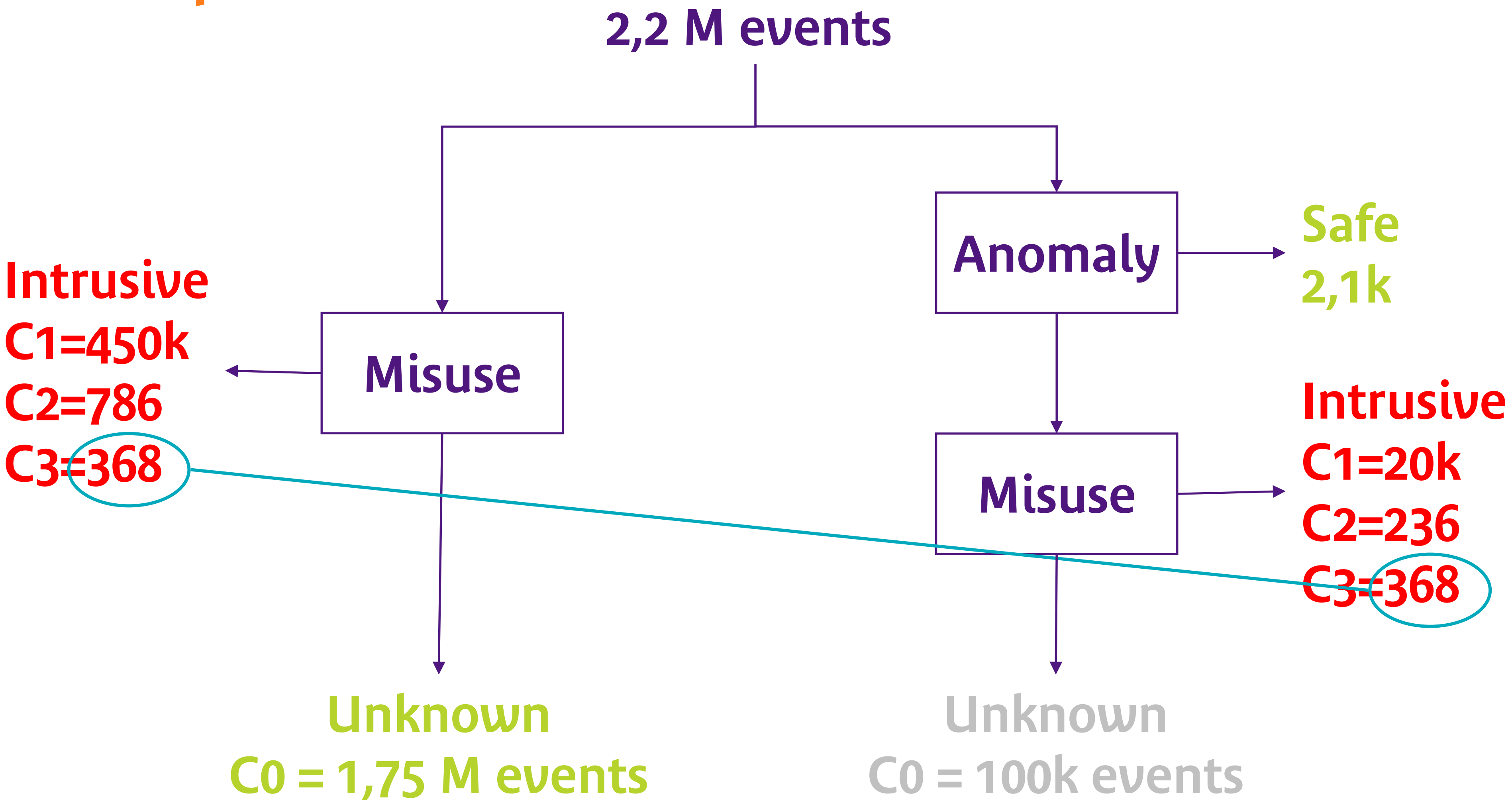
Model evaluation

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

(Unrestricted)

Example results



(Unrestricted)

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

(Unrestricted)

What is improved ?

- **False alarm rate divided by 20**
  - C1 from 450k to 20k
  - C2 from 786 to 238
- **Events analyzed by the WebAnalyzer divided by 20**
  - from 2.2M to 120k
- **Unknown events can now be investigated**
  - from 1.75M to 100k

(Unrestricted)

Discussion about such an approach

Issues (related to behaviour model)

- Can miss attacks with parameters value
- Manual construction and updates of the behavior

Advantages

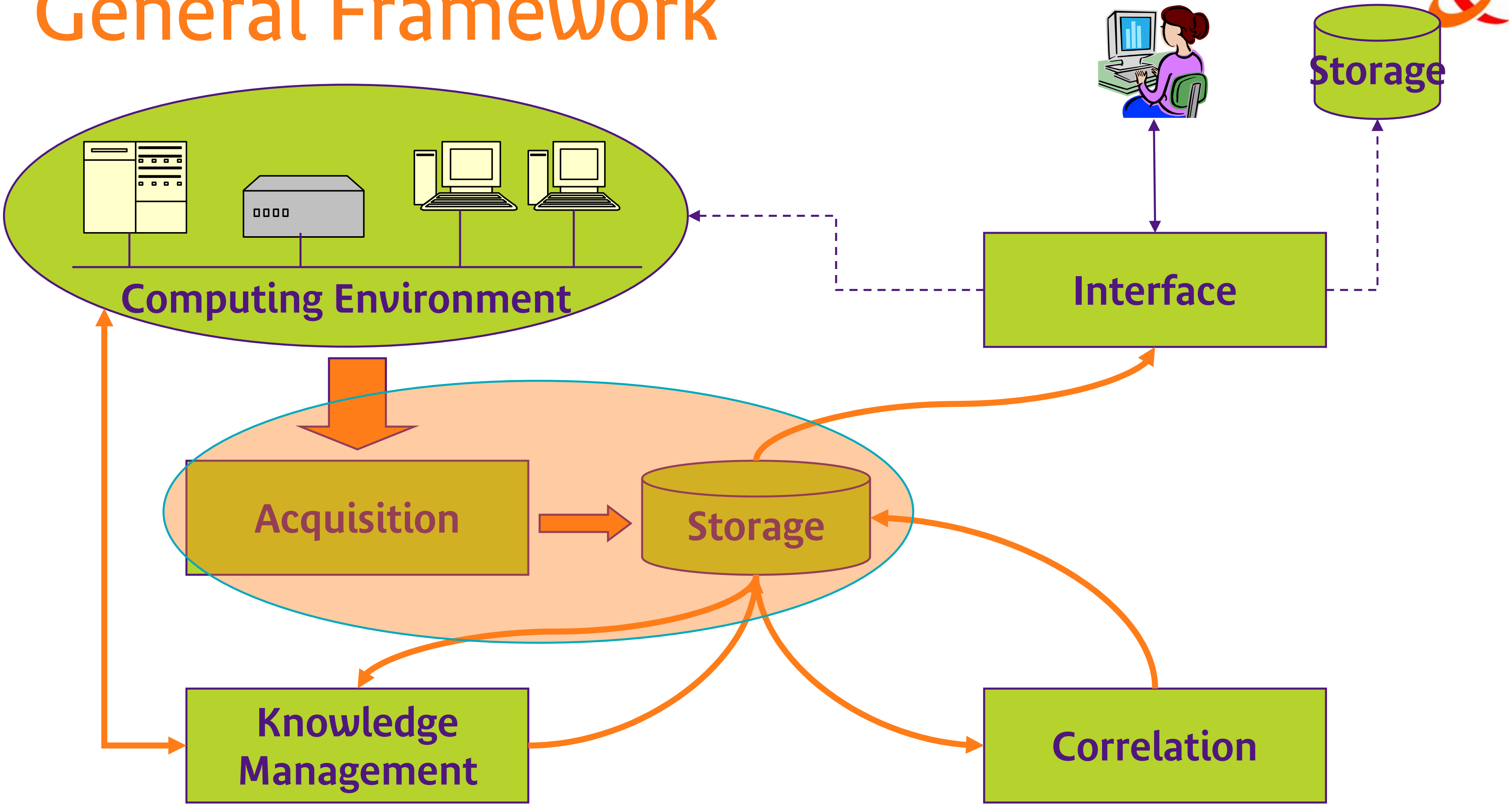
- Decreases false positive rate
- Saves time for misuse detection
  - Fine diagnosis
- Could detect new attacks

Combination of misuse and anomaly detection appearing

- But no ordered sequence of actions
- No major technological breakthrough on anomaly detection

(Unrestricted)

General Framework

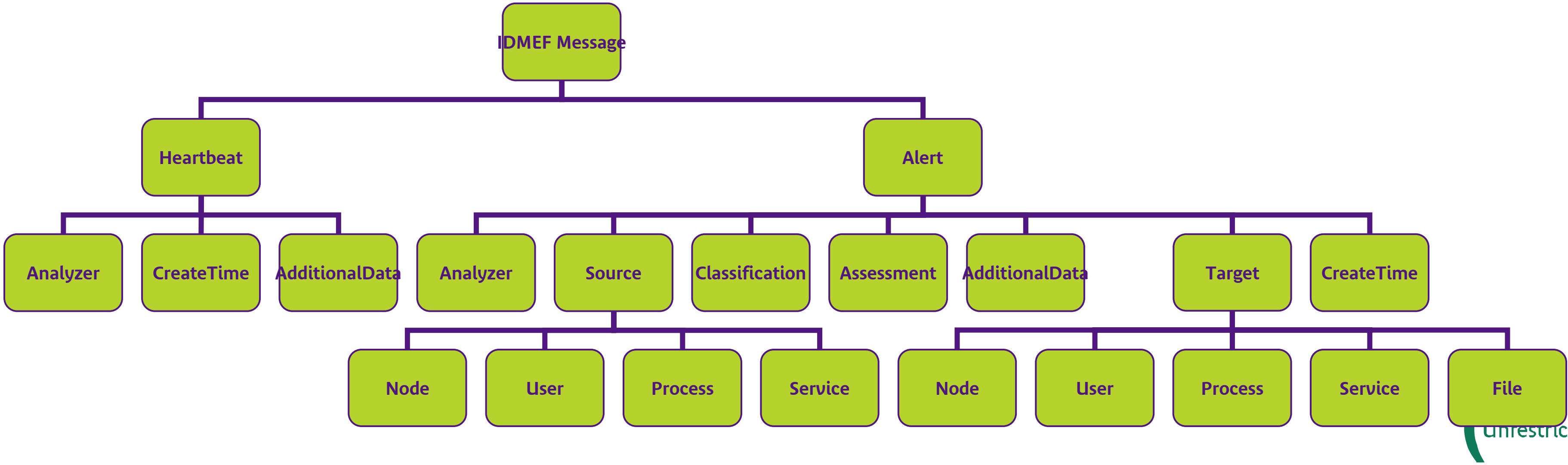


(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Intrusion Detection Message Exchange Format

- **Requirements : definition of terms**
- **Intrusion Detection eXchange Protocol**
  - Define transport between analyzer and manager
- **IDMEF**
  - Define message format between analyzer and manager
  - UML design, XML schema implementation



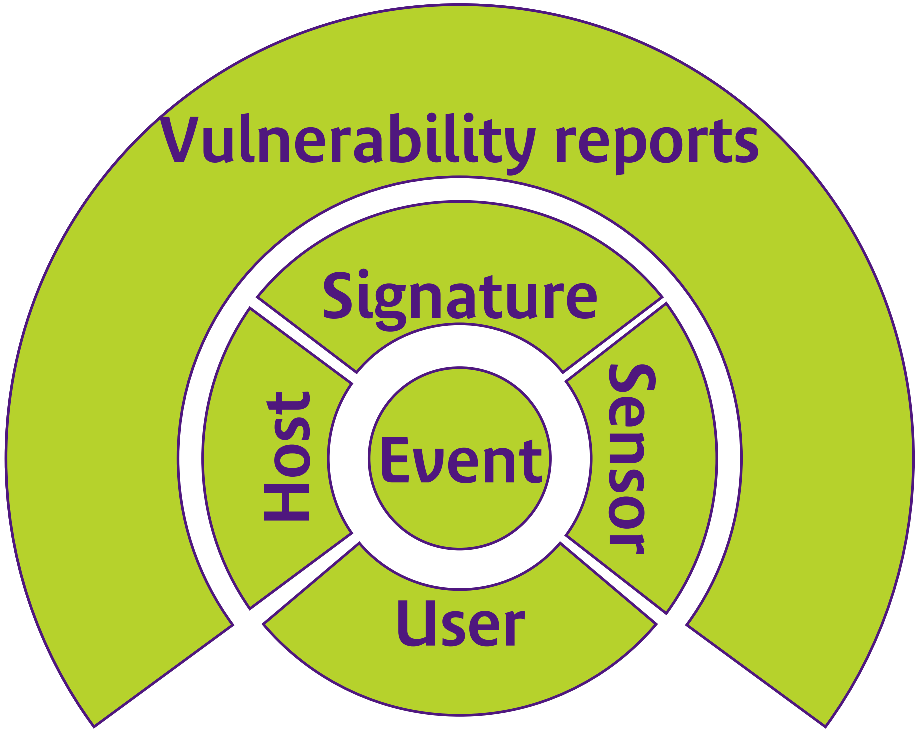
- IDMEF Message
  - Heartbeat
    - Analyzer
    - CreateTime
    - AdditionalData
  - Alert
    - Analyzer
    - Source
      - Node
      - User
      - Process
      - Service
    - Classification
    - Assessment
    - AdditionalData
    - Target
      - Node
      - User
      - Process
      - Service
      - File
    - CreateTime

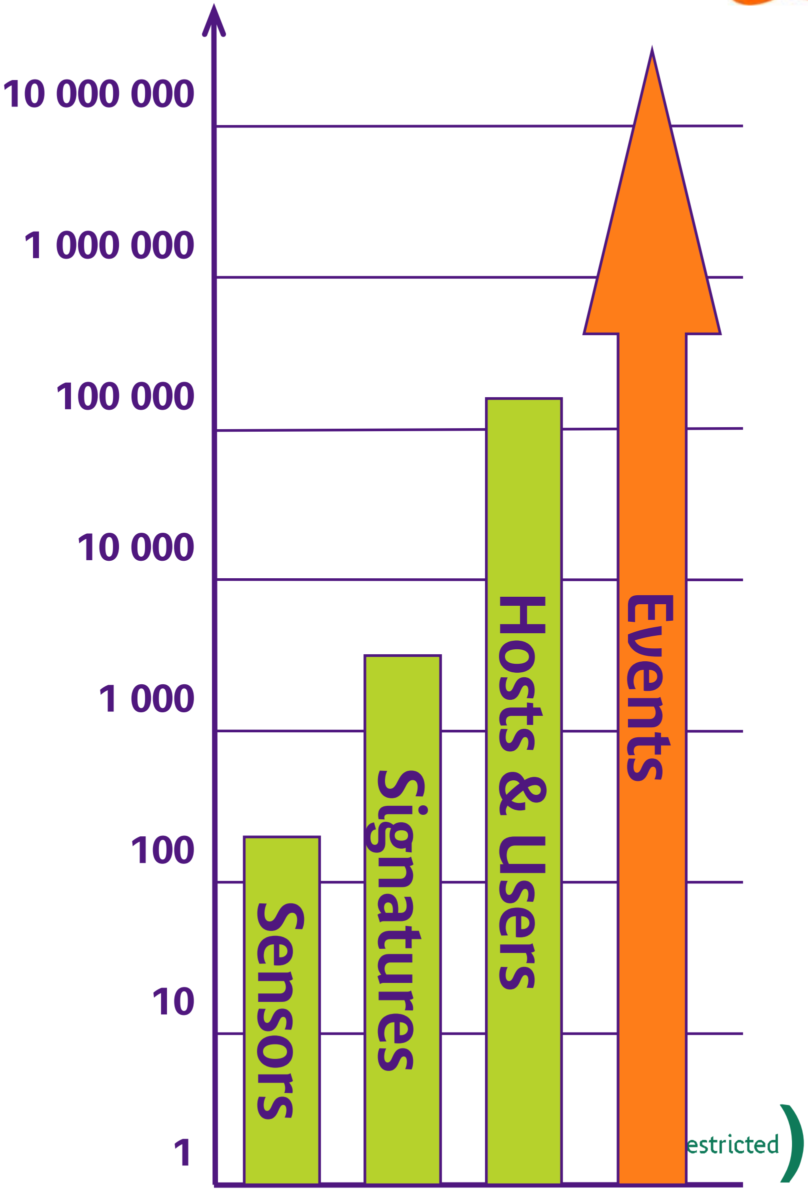
Alert representation

- **Intrusion Detection Message Exchange Format (-14)**
  - XML messages describing alerts
  - Long in the making, used mostly in research projects
  - Competing standards: SDEE, AVDL (OASIS)
  - Forward : converging alert content
- **Alert storage: not IDMEF**
  - Proper management of the ident keyword
  - Contextual data (inventory, vulnerability assessment)
  - Correlation data
- **Types of data**
  - Configuration information
  - Highly dynamic: alerts
  - Moderately dynamic: inventory, vulnerability assessment
  - Private to certain processes

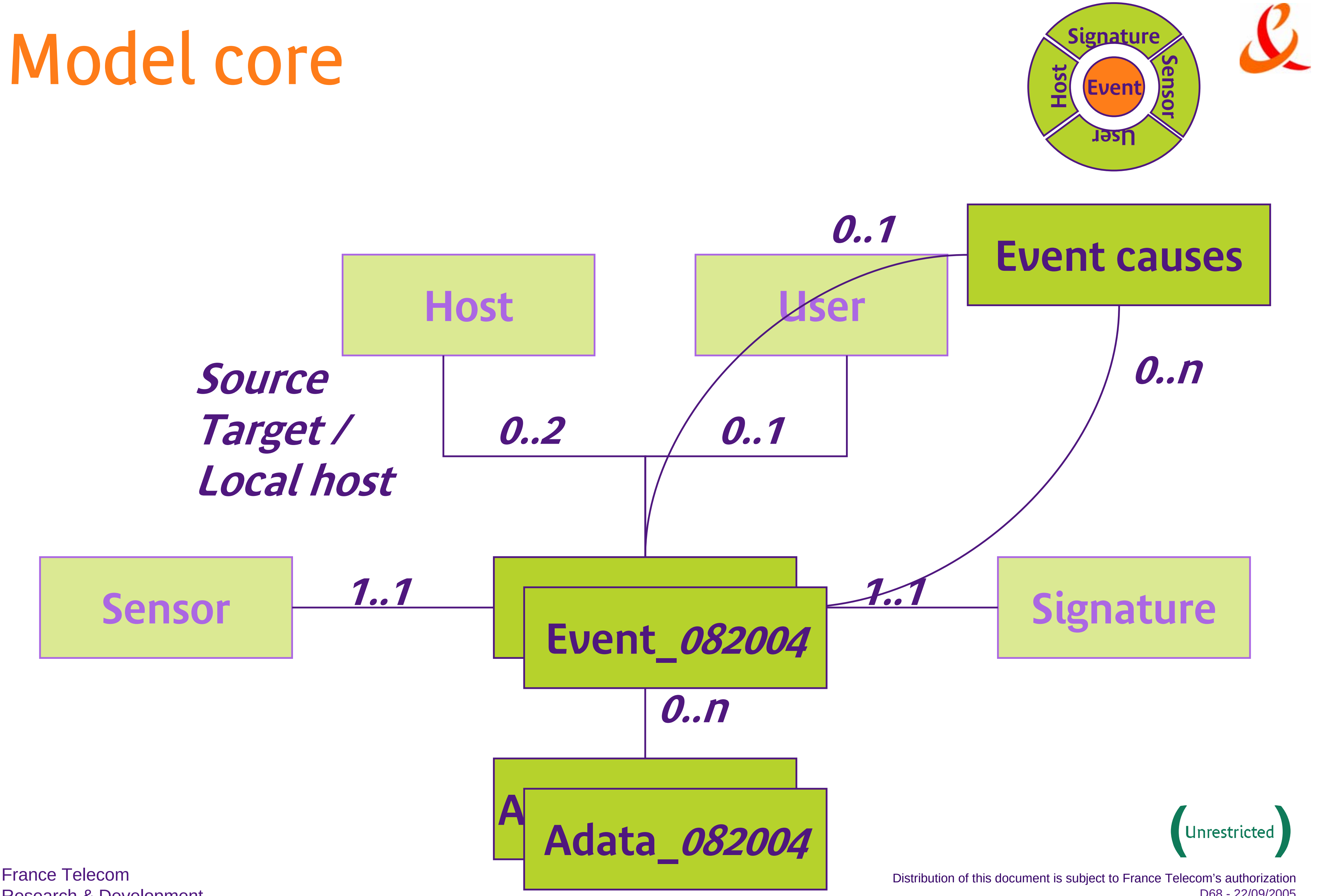
(Unrestricted)

Circular model





Model core


Signature
Sensor
User
Host
Event
Event causes
0..1
Host
User
0..n
Source
Target /
Local host
0..2
0..1
Sensor
1..1
Event_082004
1..1
Signature
0..n
A
Adata_082004

What is an event

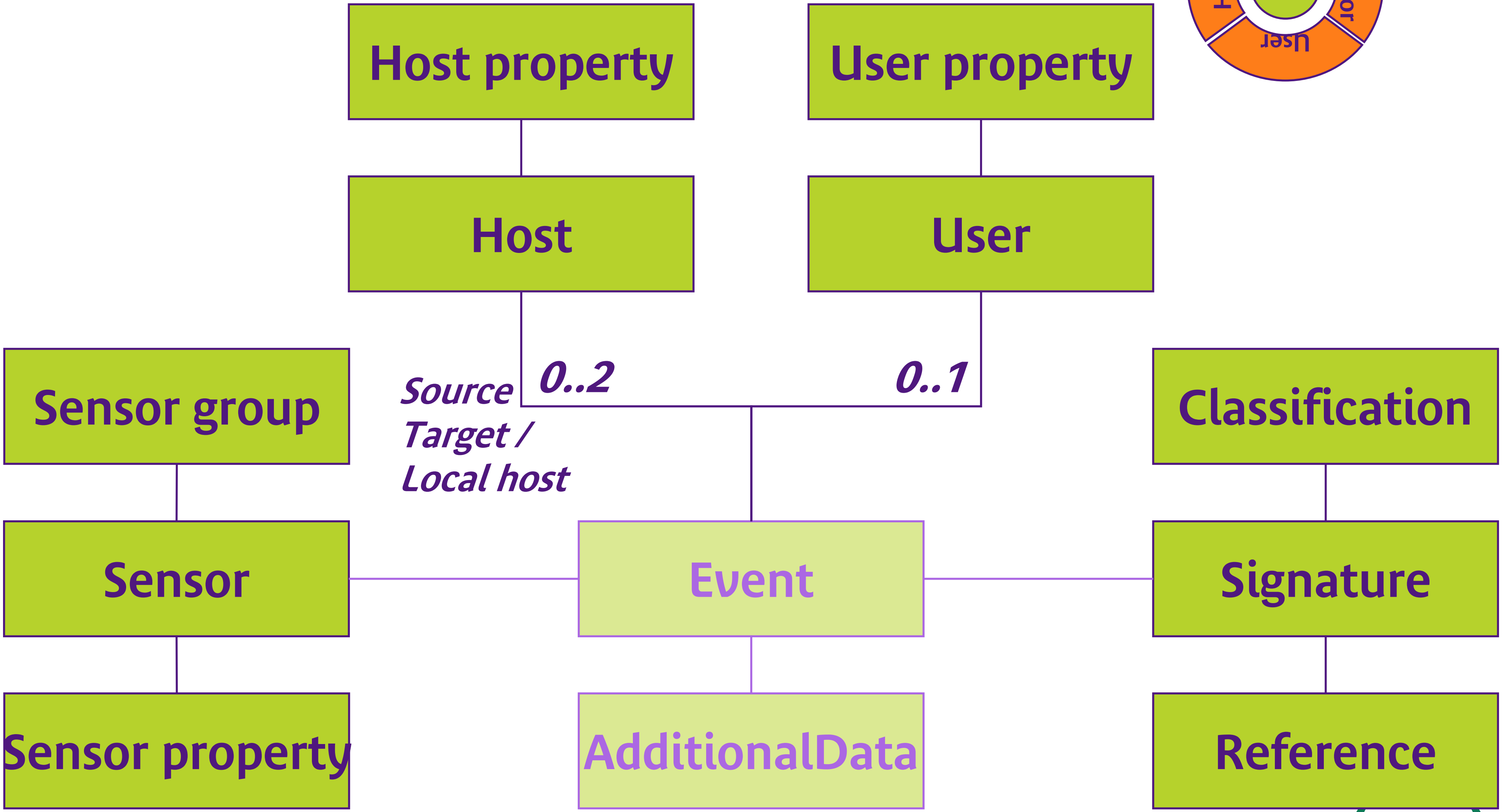
Signature
Host
Event
Sensor
User

- **3 key fields**
  - Who : the event creator
  - What: signature / classification / references / assessment
  - When : timestamp
- **Fusion between network and system data**
- **Foreign key to machines and users**
  - Hosts: sources, destinations or local
  - Ports, protocol type
  - Users
- **IDMEF-like additional data**
  - CASCADE relationship between the two tables

(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Contextual information



Host property

User property

Host

User

Source
Target /
Local host

0..2

0..1

Sensor group

Classification

Sensor

Event

Signature

Sensor property

AdditionalData

Reference

Mandatory contextual information

- **Sensor**
  - Identifier
  - Machine
  - Interface (NIDS)
  - Description
- **Sensor group**
  - Geographic aggregate
  - Functional aggregate
  - Access control or filter
- **Sensor properties**

- **Signature**
  - Meaning of the event
- **Associated with references**
  - External documentation
    - CVE
    - BID
  - Incident resolution guidelines
- **Associated with classifications**
  - Similar to properties

(Unrestricted)

Optional contextual information

Hosts

- Representation of all possible equipments
  - Servers, proxies,
  - Routers, firewall, BAS, …
- Representation with multiple addresses (MAC, IP, hostname)
- Keyed address to resolve conflicting, changing info
- Properties (10-100 / host)
  - Geographic information
  - Organizational information
  - Vulnerabilities

Users

- More difficult than hosts
- Properties

(Unrestricted)

Privileges management

Specific to each correlation process

Access by all enrichment processes

	Event	Context	Private	Causes
Acquisition	Insert	Read	-	-
Enrichment	Read	Read Insert Update Delete	-	Read
Correlation	Read Update	Read	Read Insert Update Delete	Read Insert Update
Presentation	Read	Read	- (Read ?)	-
Maintenance	Read Delete	Read Delete	-	-

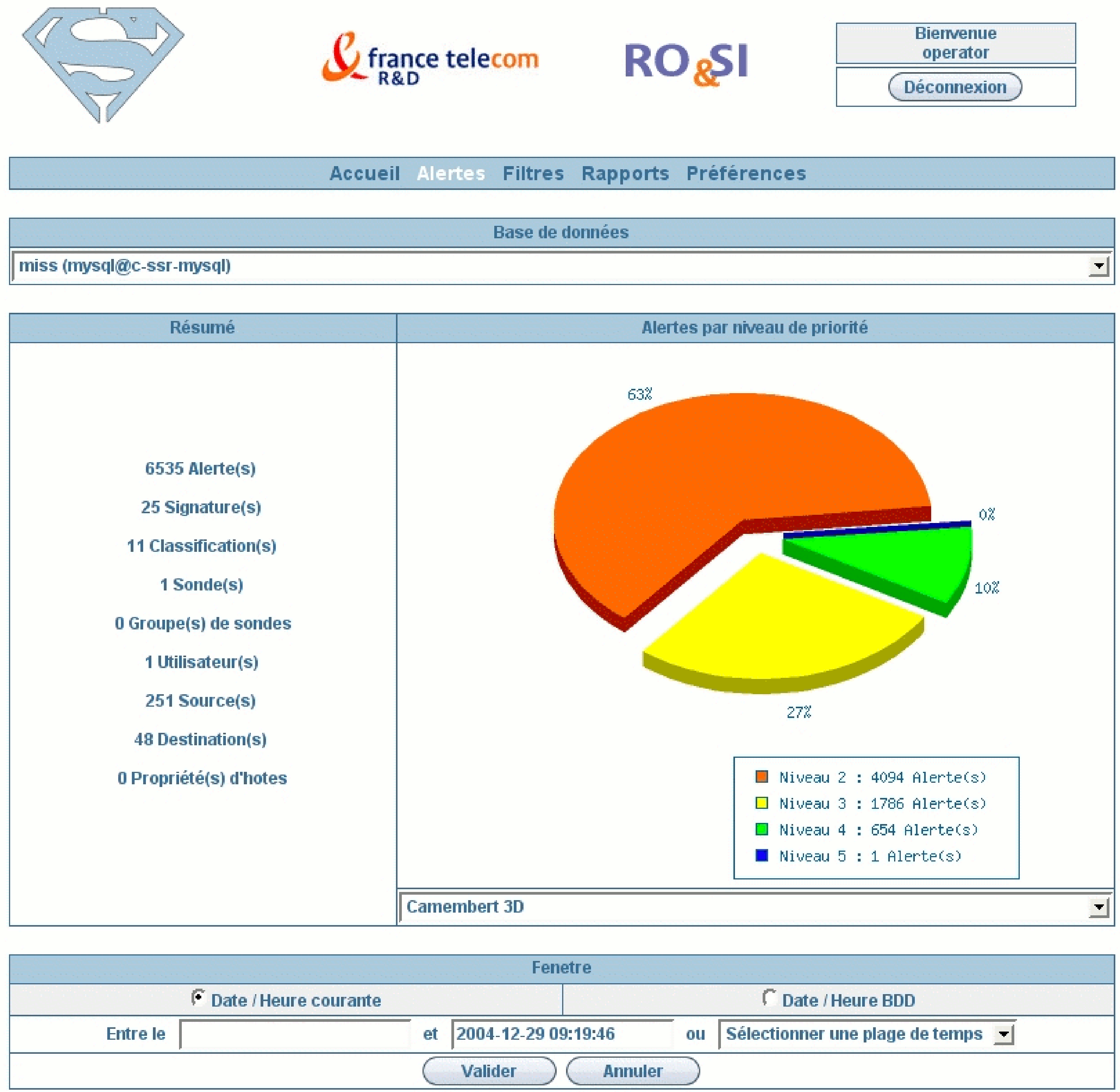
(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Advantages of structured model

- **Effective event storage**
  - System and network
  - Unification of object representations
- **Interesting properties for**
  - Data protection
  - Performance
- **Alert correlation supported by**
  - Additional alerts
  - Parent-child causal relationship
  - Hide uninteresting phenomena
- **Independent access to event information by multiple processes**

(Unrestricted)



Bienvenue operator

Déconnexion

Accueil Alertes Filtres Rapports Préférences

Base de données

miss (mysql@c-ssr-mysql)

Résumé	Alertes par niveau de priorité
6535 Alerte(s) 25 Signature(s) 11 Classification(s) 1 Sonde(s) 0 Groupe(s) de sondes 1 Utilisateur(s) 251 Source(s) 48 Destination(s) 0 Propriété(s) d'hotes	🟧 Niveau 2 : 4094 Alerte(s) 🟨 Niveau 3 : 1786 Alerte(s) 🟩 Niveau 4 : 654 Alerte(s) 🟦 Niveau 5 : 1 Alerte(s) Camembert 3D

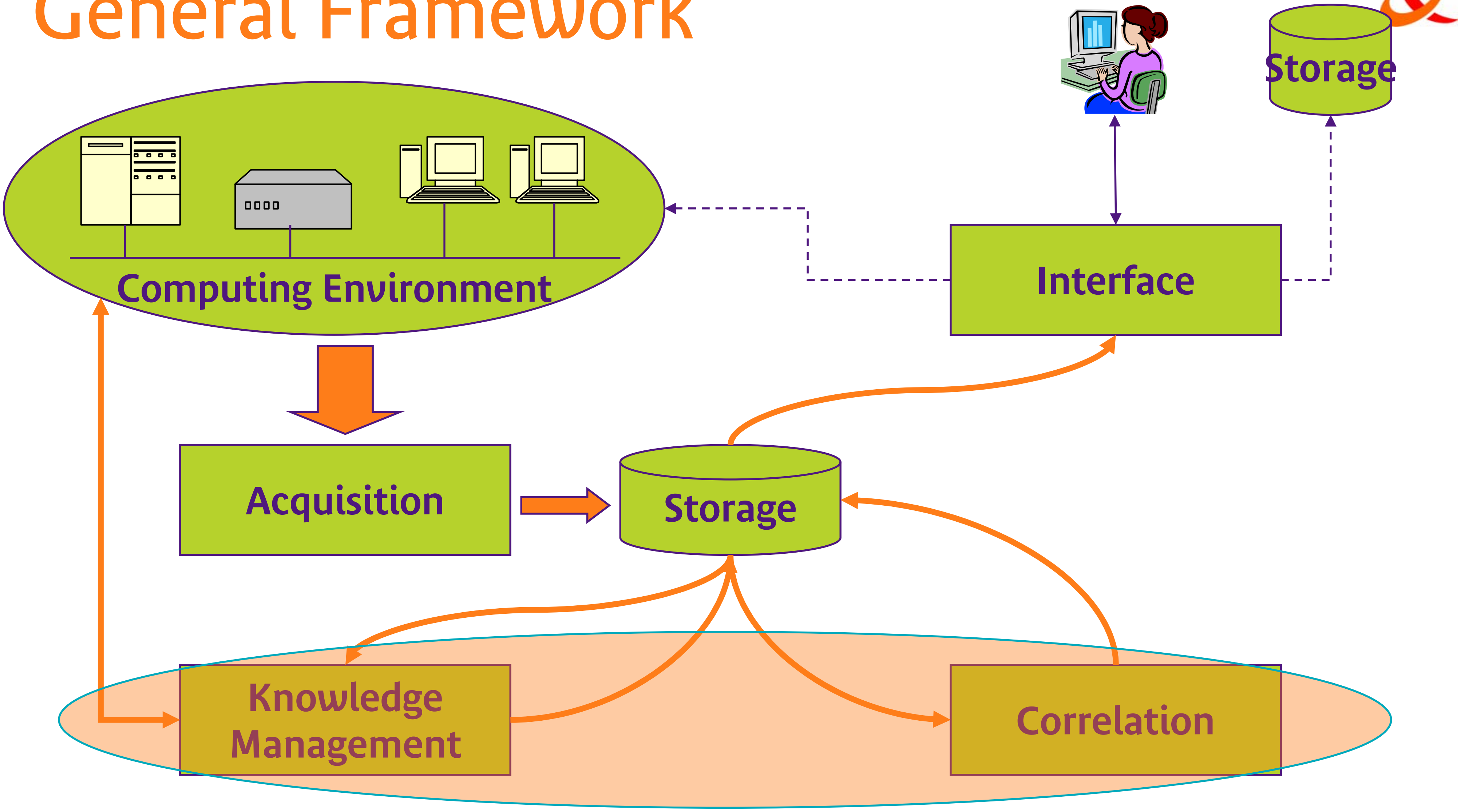
Fenetre

● Date / Heure courante ○ Date / Heure BDD

Entre le ____ et 2004-12-29 09:19:46 ou Sélectionner une plage de temps

Valider Annuler

General Framework



Why correlation ?

- **Improve detection coverage**
  - Sensors in multiple locations
  - Multiple sensors with different technologies
  - Need for a "resolver" (failed)
- **Massive amount of information (too many alerts !)**
  - Improve understanding of alerts automatically
  - Eliminate false alarms
- **Very successful area of research**
  - Pioneered in 1998, mainstream in 2000
  - Triggered standardization in the intrusion detection world
    - Along with DARPA requesting comparative evaluation
  - See slide Alert management products for the reason

(Unrestricted)

Correlation methods

- **Explicit correlation**
  - Relationships between alerts described by correlation mechanism
- **Implicit correlation**
  - Use of statistical techniques to group alerts sharing the same characteristics
    - Hopefully related
- **Semi-explicit correlation**
  - Explicit relationships
  - Hypotheses to handle missing events

(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Correlation objectives [TSI 2004]

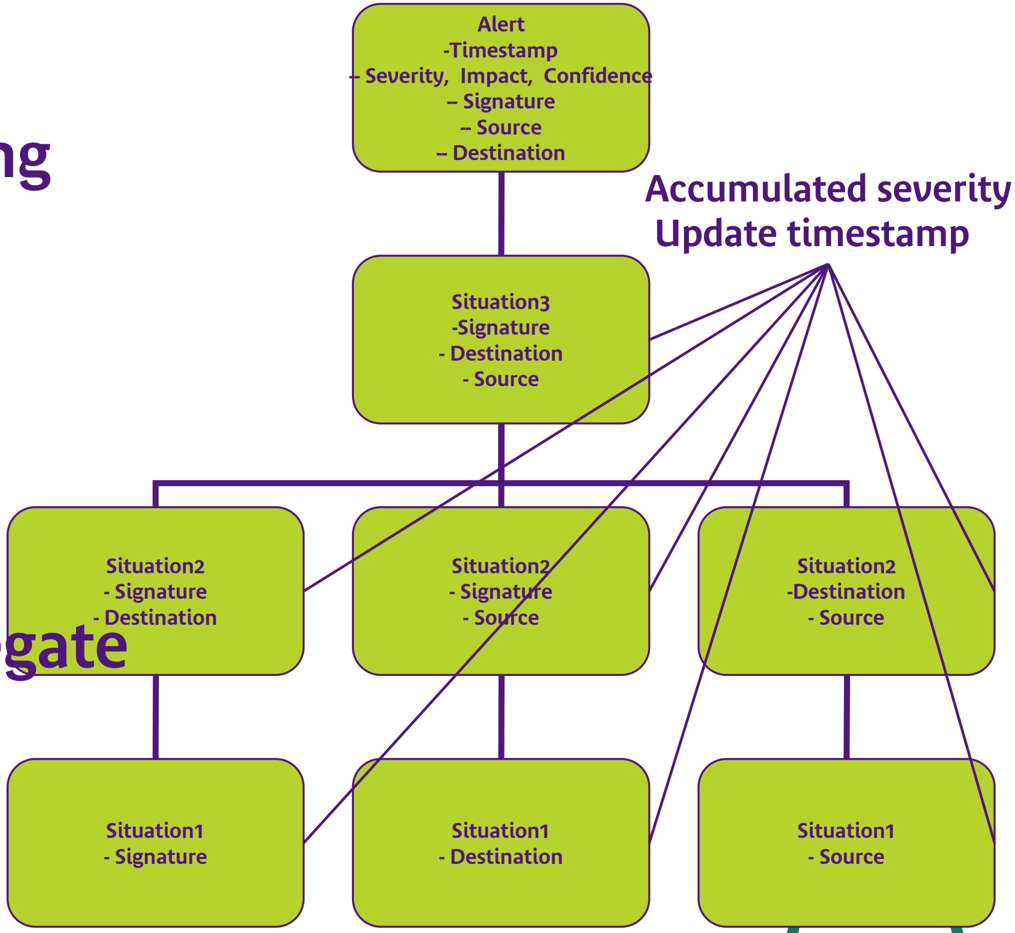
- **Reduction of alert volume**
  - Elimination
  - Fusion
  - Agregation
  - Synthesis
- **Semantic improvement**
  - Pertinence of the attack with respect to the attacked system
  - Intention of the attacker
  - Response of the application
- **Threat assessment and tracking**
  - Deep attacks
  - Wide attacks (worms)

(Unrestricted)

Correlation with TRM (Debar, Wespi)

- Duplicates removal
- Threshold based storm handling

- Project alerts on 3 axes
  - Message (signature)
  - Destination (victim)
  - Source (attacker)
- Volumetric, incremental aggregate
  - Each event contributes
  - Wider groupings
- Difficulty : explain changes



Alert
-Timestamp
- Severity, Impact, Confidence
– Signature
– Source
– Destination

Accumulated severity
Update timestamp

Situation3
-Signature
- Destination
- Source

Situation2
- Signature
- Destination

Situation2
- Signature
- Source

Situation2
-Destination
- Source

Situation1
- Signature

Situation1
- Destination

Situation1
- Source

Unrestricted

Scenarios with CRIM (Cuppens)

- **Create post/pre relationships**
  - Pre-requisites of alert match post-results of another
  - Same context (user, machine, time)
  - Attack scenarios without scenarios
  - Missing & unobservable events "created"
- **Create meta alert with all threat characteristics**
- **Problem: assign post and pre conditions to alerts**
  - During signature writing
  - Based on references and additional external information
  - Limited
  - Tedious

(Unrestricted)

Simpler: Chronicles (Dousson, Morin)

- **No event simulation**
- **Temporal relationship**
  - Causality links
  - Explicit : synthesis
  - Follow activities
- **Interest**
  - Create relationships
  - High-volume, recurring things
    - Worms
  - Efficient (CPU-wise)
- **Difficulty**
  - Write chronicles
  - FACE: chronicle discovery

```
chronicle nimda[?attacker, ?victim]
{
  occurs((1,2),alarm[?sensor, ?, iis_code_red_ii_root_exe,
                     ?attacker,?victim], (t,t+2))
  occurs((1,4),alarm[?sensor, ?, iis_decode_bug,
                     ?attacker,?victim],(t,t+2))
  occurs((1,14),alarm[?sensor, ?, iis_cmd_exe,
                      ?attacker,?victim],(t, t+2))
  occurs((1,3),alarm[?sensor, ?, web_dot_dot,
                     ?attacker,?victim],(t,t+2))
  occurs((1,2),alarm[?sensor, ?, iis_unicode,
                     ?attacker,?victim],(t,t+2))
  occurs((1,1),alarm[?sensor, ?, iis_unicode2,
                     ?attacker,?victim],(t,t+2))
  occurs((1,1),alarm[?sensor, ?, iis_unicode3,
                     ?attacker,?victim],(t,t+2))
  occurs((1,1),alarm[?sensor, ?, iis_decode_bug3,
                     ?attacker,?victim],(t,t+2))
  occurs((1,1),alarm[?sensor, ?, iis_decode_bug2,
                     ?attacker,?victim],(t,t+2))
  occurs((1,1),alarm[?sensor, ?, iis_decode_bug4,
                     ?attacker,?victim],(t,t+2))

  when recognized {
     emit event(chroniclealarm[nimda_attempt, ?attacker,
                               ?victim], t);
  }
}
```

Contextual host information

- **Knowledge of the monitored environment**
  - Vulnerability assessment
  - Inventory
- **Prioritize treatment by assessing risk**
  - Risk represented by the existence of vulnerability
  - Relationship established through equal references
  - Inventory requires assessment of version numbers
- **Information obtained through**
  - Vulnerability assessment (nessus, nmap, nikto, …)
  - Passive network observation

(Unrestricted)

Nessus report, results

```
<host name="xxx" ip="xxx"/>
<date>
          <start>Fri Sep  5 11:03:55 2003</start>
          <end>Fri Sep  5 11:06:40 2003</end>
</date>
<ports>
<port protocol="udp" portid="177">
          <service name="xdmcp" method="nessus" conf="3" />
          <information>
                    <severity>Security Warning</severity>
                    <id>10891</id>
                    <data>    The remote host is running XDMCP.
                              ..
                              Risk factor : Medium
                              Solution : Disable XDMCP
                    </data>
          </information>
</port>
<port protocol="tcp" portid="80">
          <service name="www" method="nessus" conf="3" />
          <information>
                    <severity>Security Hole</severity>
                    <id>11030</id>
                    <data>    The remote host appears to be vulnerable to the Apache
                              Web Server Chunk Handling Vulnerability.
                              …
                              *** Note : as safe checks are enabled, Nessus solely relied on
                                  the banner to issue this  alert
                              Solution : Upgrade to version 1.3.26 or 2.0.39 or newer
                              See also : http://httpd.apache.org/info/security_bulletin_20020617.txt
                              http://httpd.apache.org/info/security_bulletin_20020620.txt
                              Risk factor : High
                              CVE : CVE-2002-0392
                              BID : 5033
                    </data>
          </information>
```

(Unrestricted)

Technical problems of VA

Undesired additional traffic

- Performance issue
- Side effects on tested systems
  - Destructive checks

Timeliness of vulnerability information

- Since when ?
  - The last test
  - Unless checks where updater
- Was the vulnerability exploited ?
  - Response through intrusion detection systems

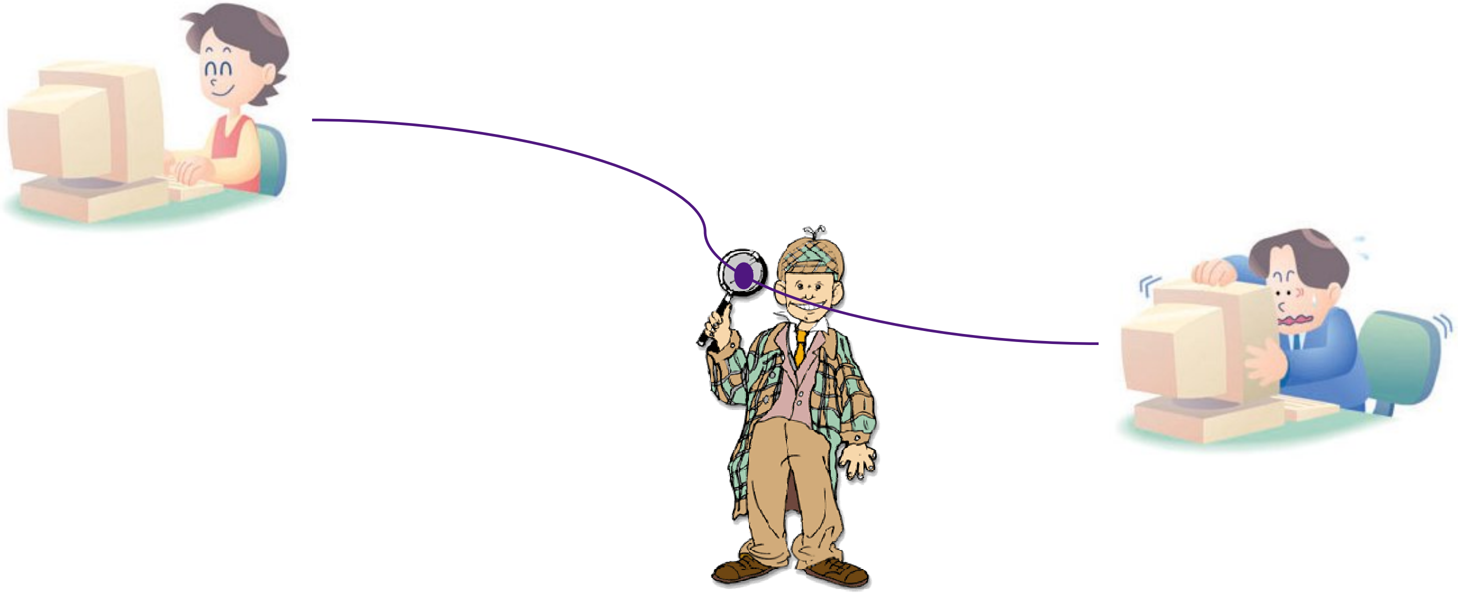
Quality of information

- Only provides server side information
- Related to accessibility

(Unrestricted)

Passive network cartography

- **Induce system characteristics from traffic observation**
  - Quite close to NIDS



Unrestricted

State of the art

- **Relatively few publications on the subject**
  - Dayioglu [ISCIS01] : basis
  - Webster *et al* : issues and experiments
  - Lippman *et al* [Rump session RAID 2004]
- **Commercial tools**
  - RNA (Real Time Network Analysis)
  - NeVO (Network Vulnerability Observer)
  - Limited capabilities
  - Quite expensive (10000$ / sensor)
- **Open source tools**
  - Ettercap
  - Limited capabilities

(Unrestricted)

Example of traffic analysis

Request `http://intranoo/`

intranoo

```
GET / HTTP/1.1\r\n
Accept: application/vnd.ms-powerpoint, ...
Accept-Language: fr\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: intranoo\r\n
Connection: Keep-Alive\r\n
```

Server name

Browser identity

Client OS

… with variations …

Request http://intranoo/

intranoo

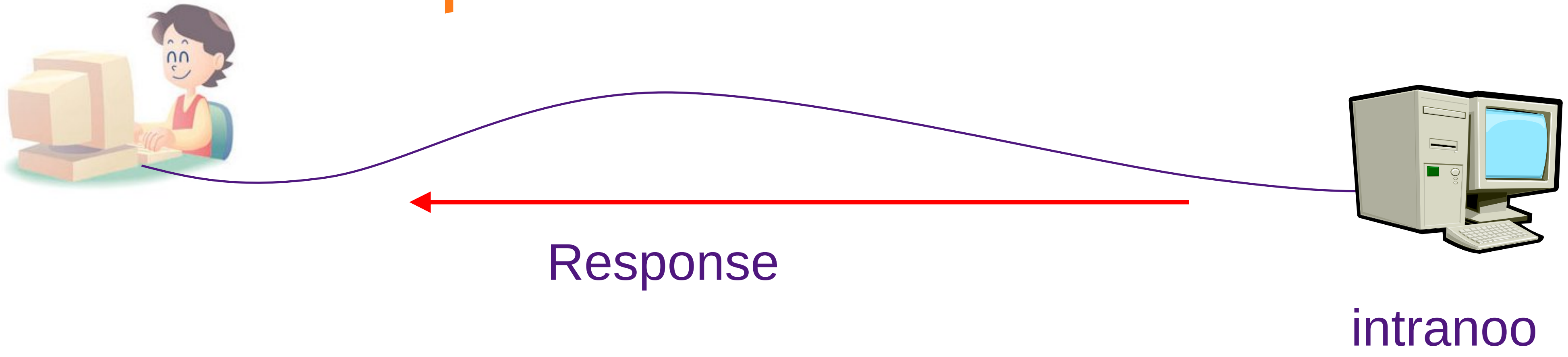
```
GET / HTTP/1.1\r\n
Request Method: GET
Host: intranoo.francetelecom.fr\r\n
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7)
Gecko/20040803 Firefox/0.9.3\r\n
Accept: text/xml,applicat...
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
```

Server name

Browser identity

Client OS

… and response



```
HTTP/1.1 200 OK\r\n
Response Code: 200
Date: Wed, 25 Aug 2004 08:53:50 GMT\r\n
Server: Apache\r\n
Last-Modified: Tue, 11 May 2004 07:46:20 GMT\r\n
Accept-Ranges: bytes\r\n
Content-Length: 300\r\n
Keep-Alive: timeout=15, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html\r\n
```

Server type

Advantages of passive cartography

- **Update "as soon as necessary"**
  - As soon as a resource is used by its environment
- **Safe and non-disturbing**
  - No traffic generated
- **Simple to deploy**
  - But need to find the "right" place
- **Discovery of client applications**

(Unrestricted)

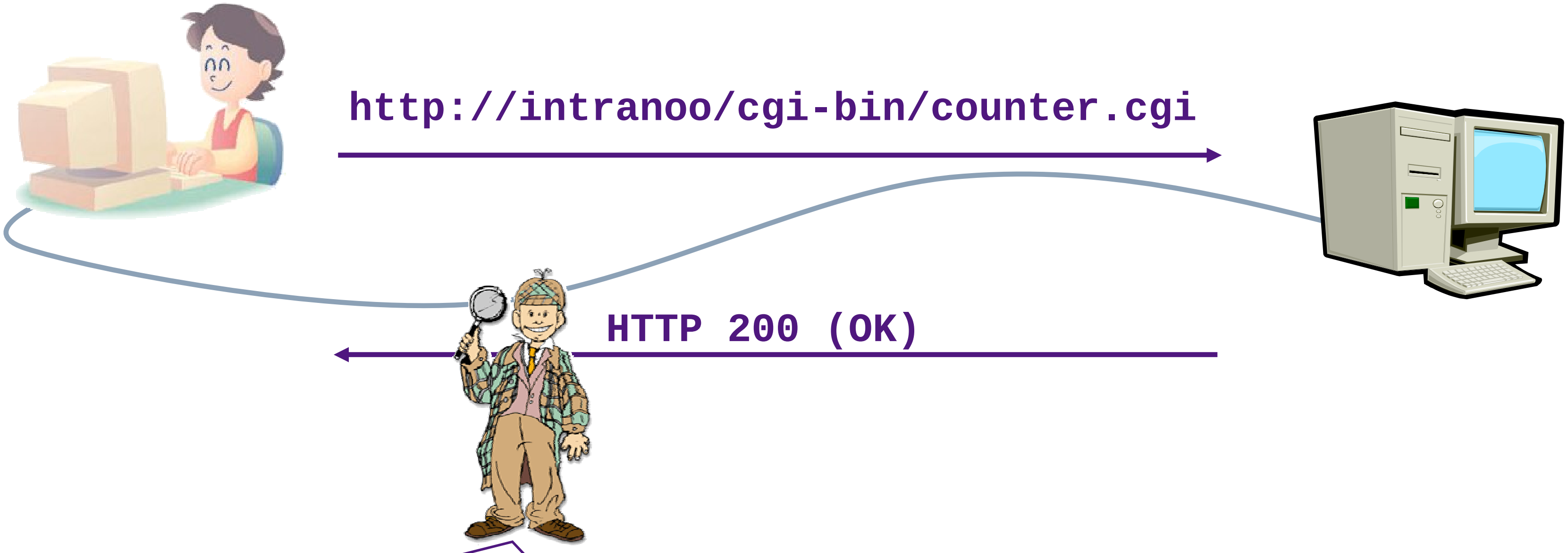
Additional interesting functions

- **Stateful analysis**
- **Protocol decodes**
  - Obtain precise information
- **Basis of NIDS/NPDS today**

- **Intelligent stuff**
  - Negative properties
    - *Avoid closed world hypothesis*
    - *Discover the absence of properties (e.g. non-installed software)*
  - Property inference
    - Deduce unobserved system characteristics
    - Confirm or infirm observations
    - Detect incoherencies

(Unrestricted)

Protocol decodes

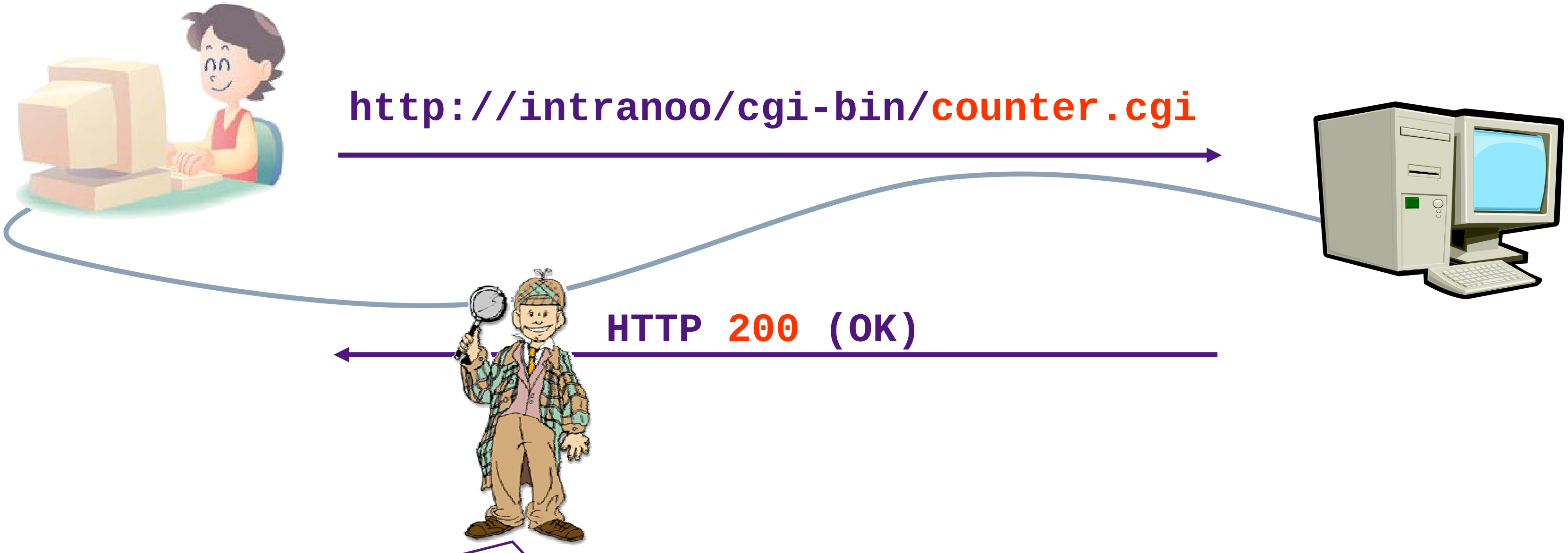


http://intranoo/cgi-bin/counter.cgi

HTTP 200 (OK)

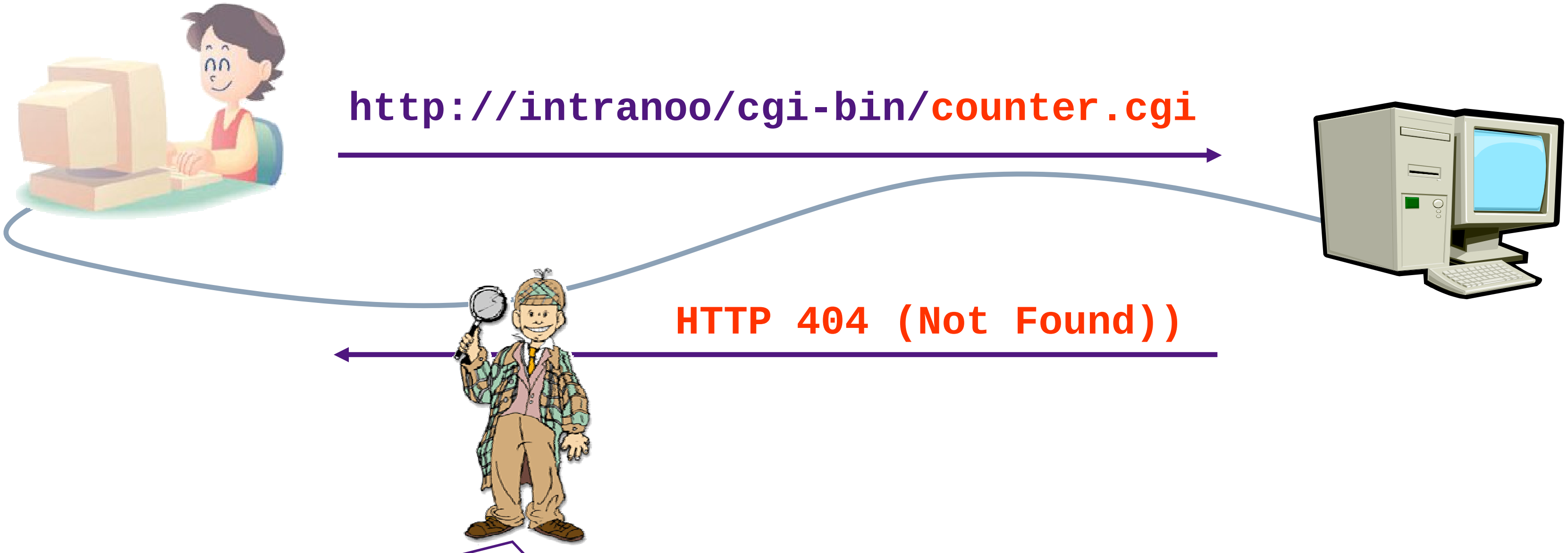
```
runs(clientid, product(Netscape, 5.3, webbrowser))
runs(clientid, product(Linux, 2.6.2, os))
runs(intranoo, product(iis, 5.1, webserver))
```

Stateful analysis



```
runs(clientid, product(Netscape, 5.3, webbrowser))
runs(clientid, product(Linux, 2.6.2, os))
runs(intranoo, product(iis, 5.1, webserver))
runs(intranoo, product(counter.cgi, 1.9, cgiapp))
```

Negative properties

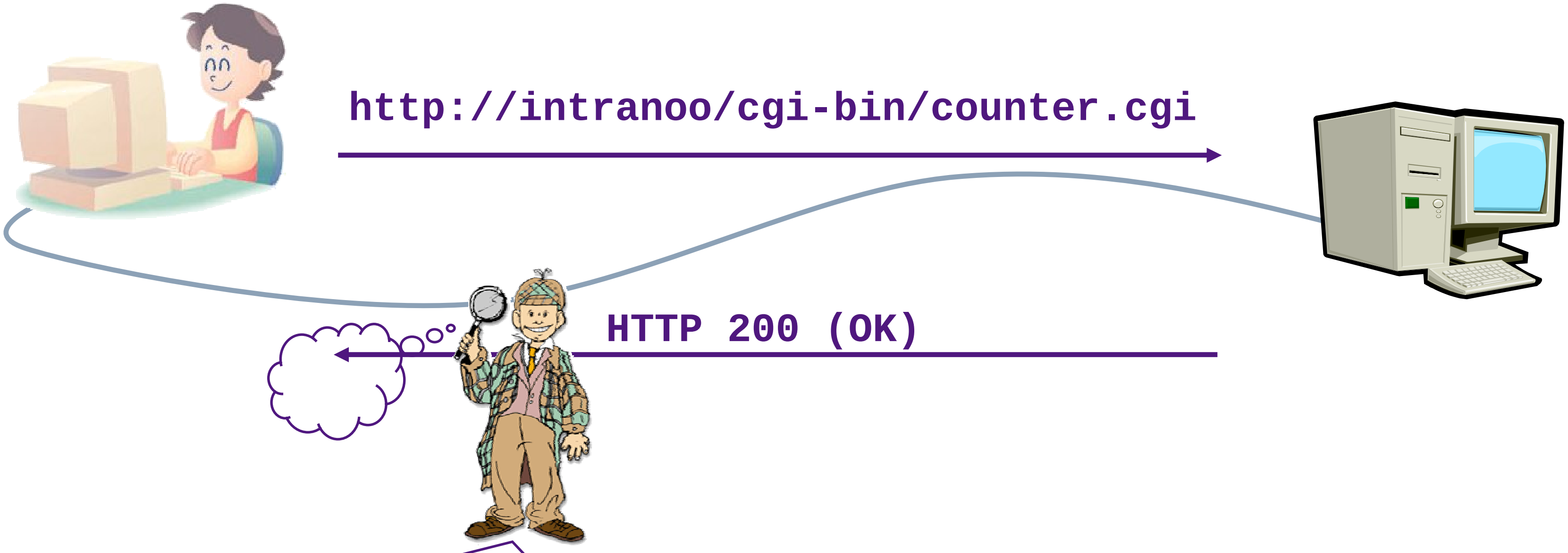


http://intranoo/cgi-bin/counter.cgi

HTTP 404 (Not Found))

```
runs(clientid, product(Netscape, 5.3, webbrowser))
runs(clientid, product(Linux, 2.6.2, os))
runs(intranoo, product(iis, 5.1, webserver))
¬runs(intranoo, product(counter.cgi, _, cgiapp))
```

Property inference



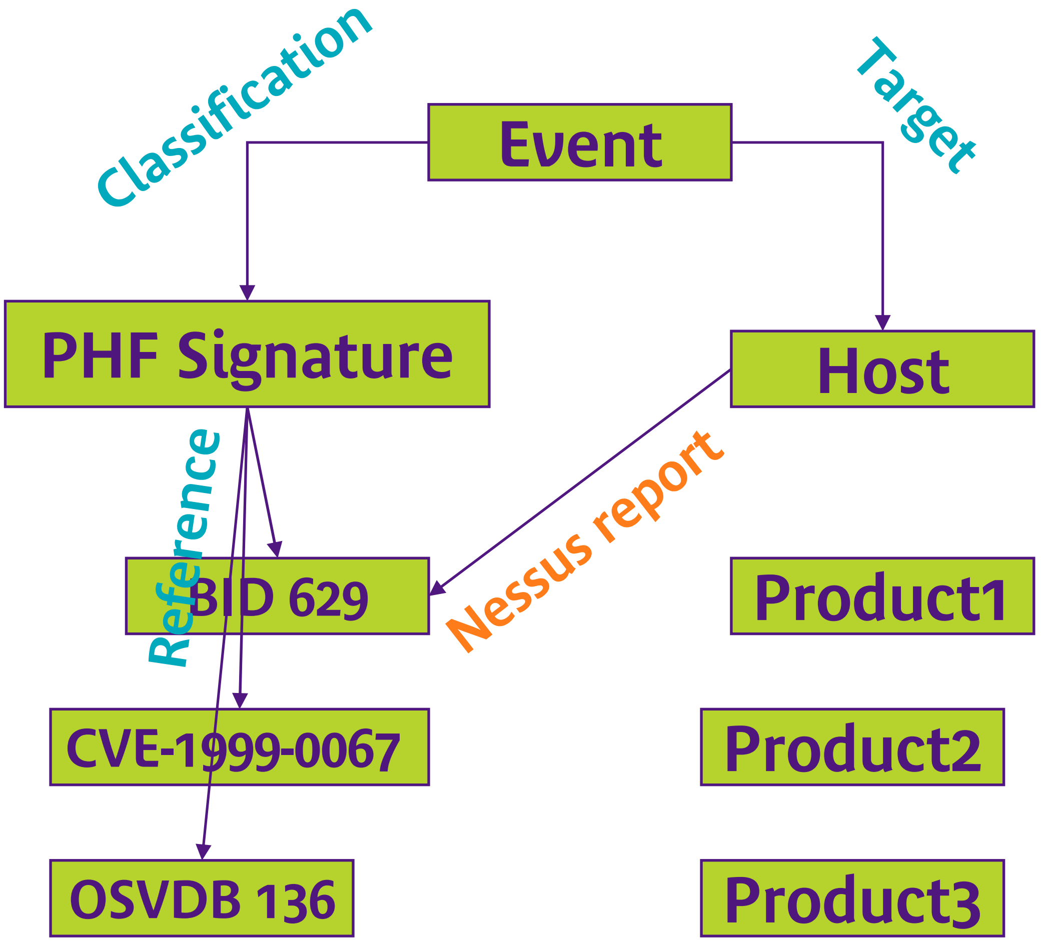
http://intranoo/cgi-bin/counter.cgi

HTTP 200 (OK)

```
runs(clientid, product(Netscape, 5.3, webbrowser))
runs(clientid, product(Linux, 2.6.2, os))
runs(intranoo, product(iis, 5.1, webserver))
runs(intranoo, product(counter.cgi, 1.9, cgiapp))
runs(intranoo, product(windows, _, os))
```

Vulnerabilities & alerts

- **Searching for relationships**
- **Alert linked to :**
  - Signature (message)
  - Host (target / source / local)
- **Signature associated with**
  - References (CVE, BID, …)
- **Host associated with**
  - References (vulnerability assessment)

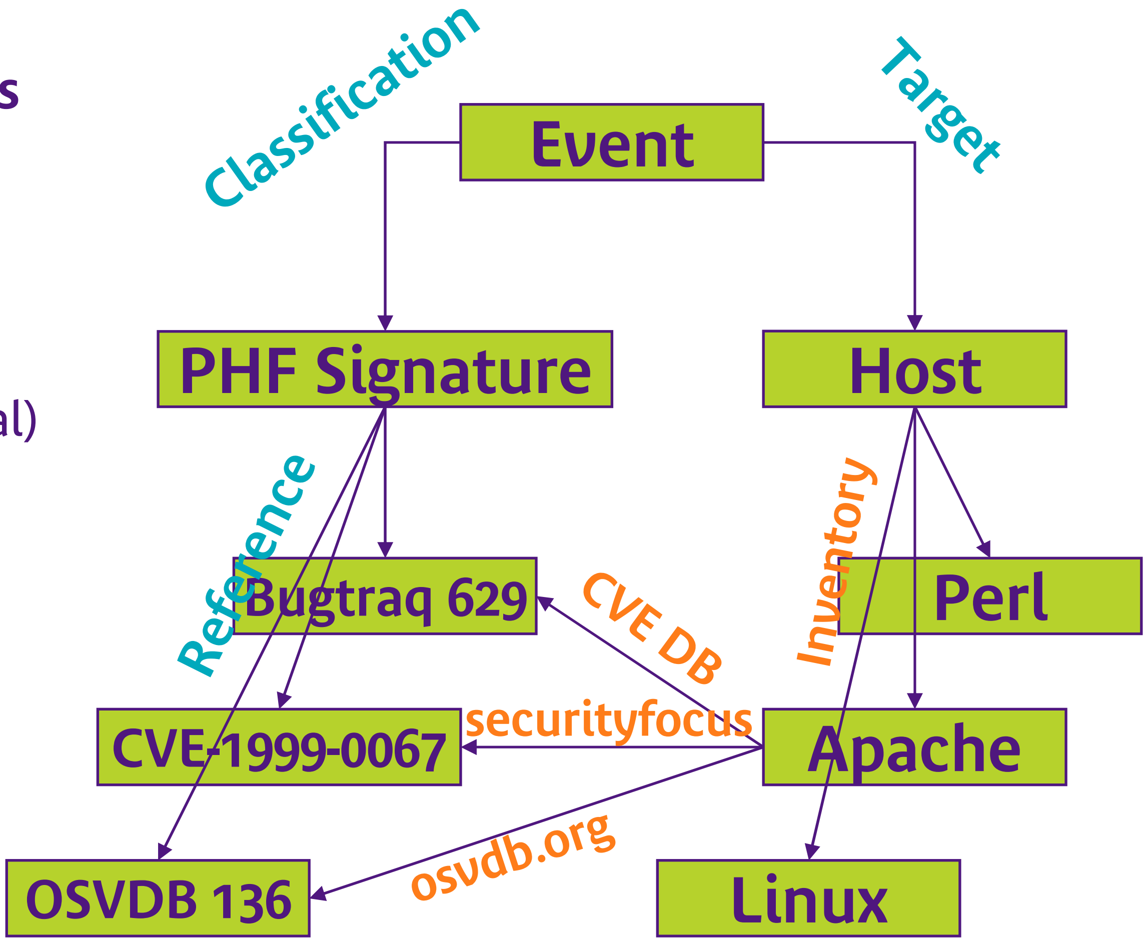


(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Inventory & alerts

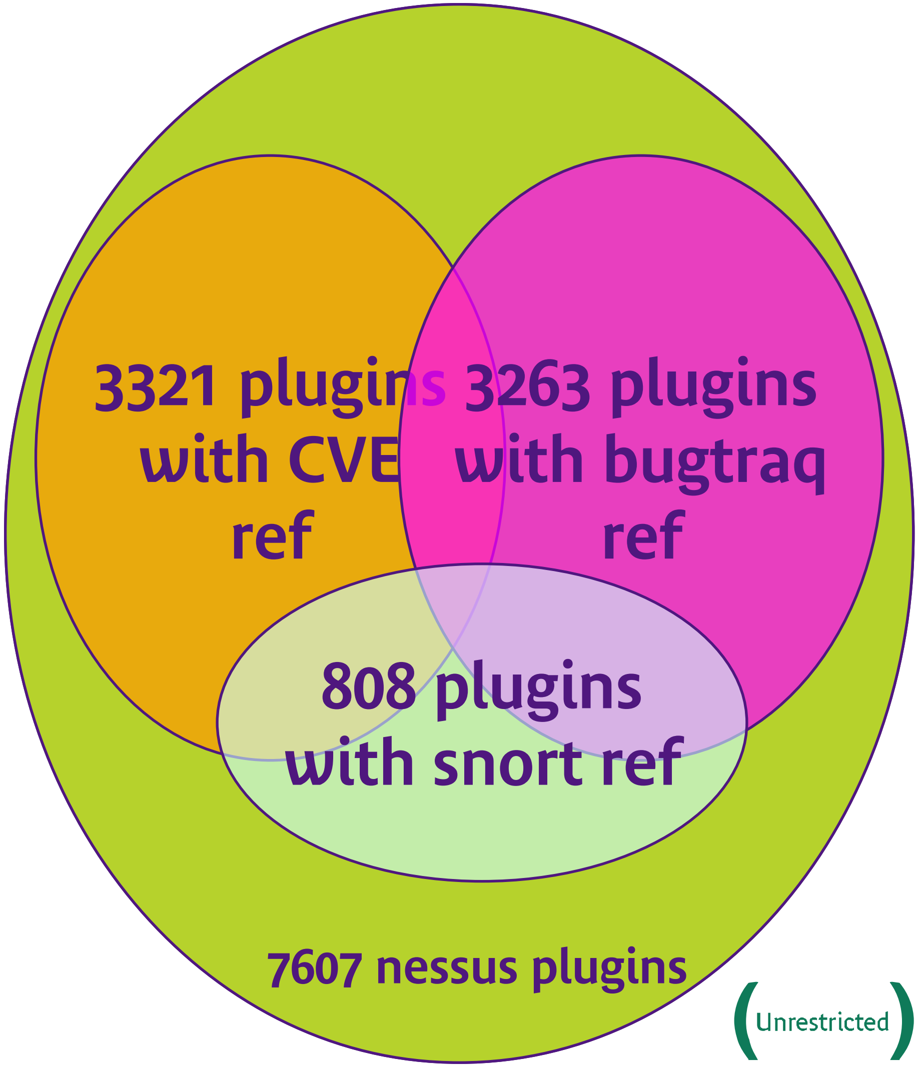
- **Searching for relationships**
- **Alert linked to :**
  - Signature (message)
  - Host (target / source / local)
- **Signature associated with**
  - Références (CVE, BID, …)
- **Host associated with**
  - Products (inventory)
- **Product associated with**
  - Références (external documentation)



(Unrestricted)

Evaluation of correlation capabilities

- **Vulnerability assessment has partial reference coverage**
  - 37.78% of plugins without usable reference
- **Intrusion detection has partial coverage**
  - 39,46% of signature without usable reference
- **Correlation capability w.r.t sigs**
  - Active (nessus): 50%
  - Passive (osvdb): 32%
  - Active then passive: 55.73%
  - Passive then active: 36,29%


3321 plugins with CVE ref
3263 plugins with bugtraq ref
808 plugins with snort ref
7607 nessus plugins

(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Issues with this kind of correlation

- **Change management**
  - Track changes in host configuration
  - Transient ports, backdoors
- **Actual testing**
  - Implicit deduction between ports and services
  - Analysis of test responses
  - Correspondence between the nessus plugins configuration and the test results

(Unrestricted)

Evaluation of test results

- Cannot reach the test objec
  - Network issue
- Cannot test the service
  - Host issue or network issue
- Test provided no useful res
  - Unspecified error code
- Test produced unexpected
  - Unexpected error code
- Test explicitly succeeded
  - I am vulnerable to the attack
- Test explicitly failed
  - I am not vulnerable to the attack

Cannot establish TCP connexion (firewall drop),

ICMP (net|h e (firewall reject, network fai host down)

TCP Port closed

HTTP Response 500

HTTP Response 401

HTTP Response 200

HTTP Response 404

(Unrestricted)

Summary

- **This is not a silver bullet, but it is a useful tool**
- **Countermeasures can be applied to about 50% of signatures**
- **Concerns only smaller proportion of alerts**
  - The most difficult alerts
- **Mistakes found in referencing during analysis**

(Unrestricted)

Statistical correlation

- **5 signatures generate 70% of the alerts**
- **Peripheral characteristics of attacks**
  - Fingerprinting (performance measurement)
  - Attack results (ICMP messages)
- **Impossible to process manually**
  - Correspond to natural background noise
  - Undesirable to drop signatures
  - Build model of background noise

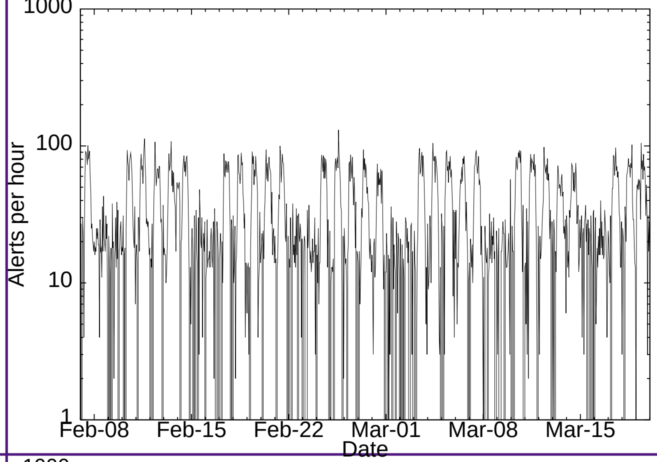
(Unrestricted)

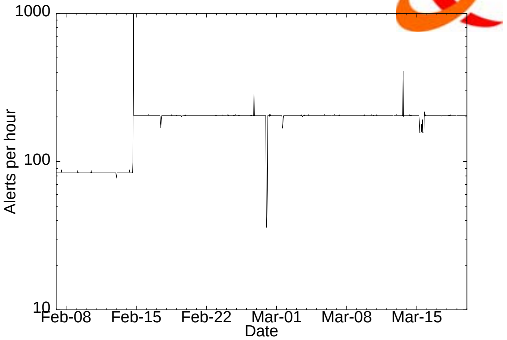
Alert types

"Well characterized alerts"
- Reference to software vuln
- Low frequency

Explicable alerts

Speedera (ICMP)



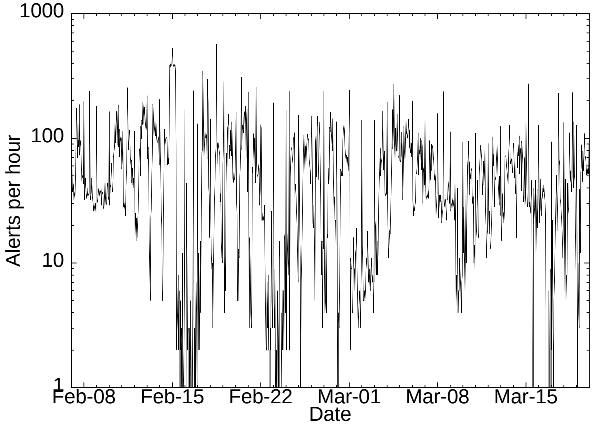


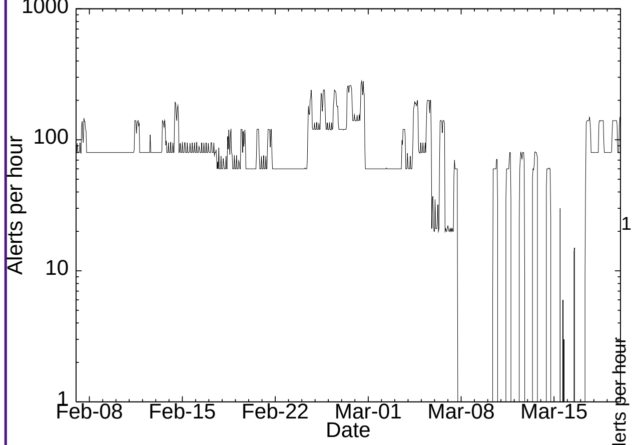
SNMP Request UDP

Low regularity

High regularity

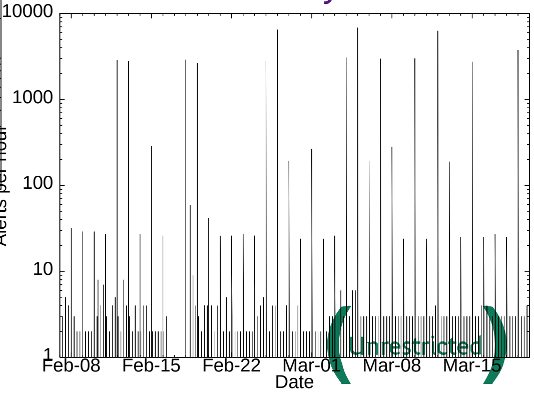
ICMP Destination unreachable
(communication administratively prohibited)





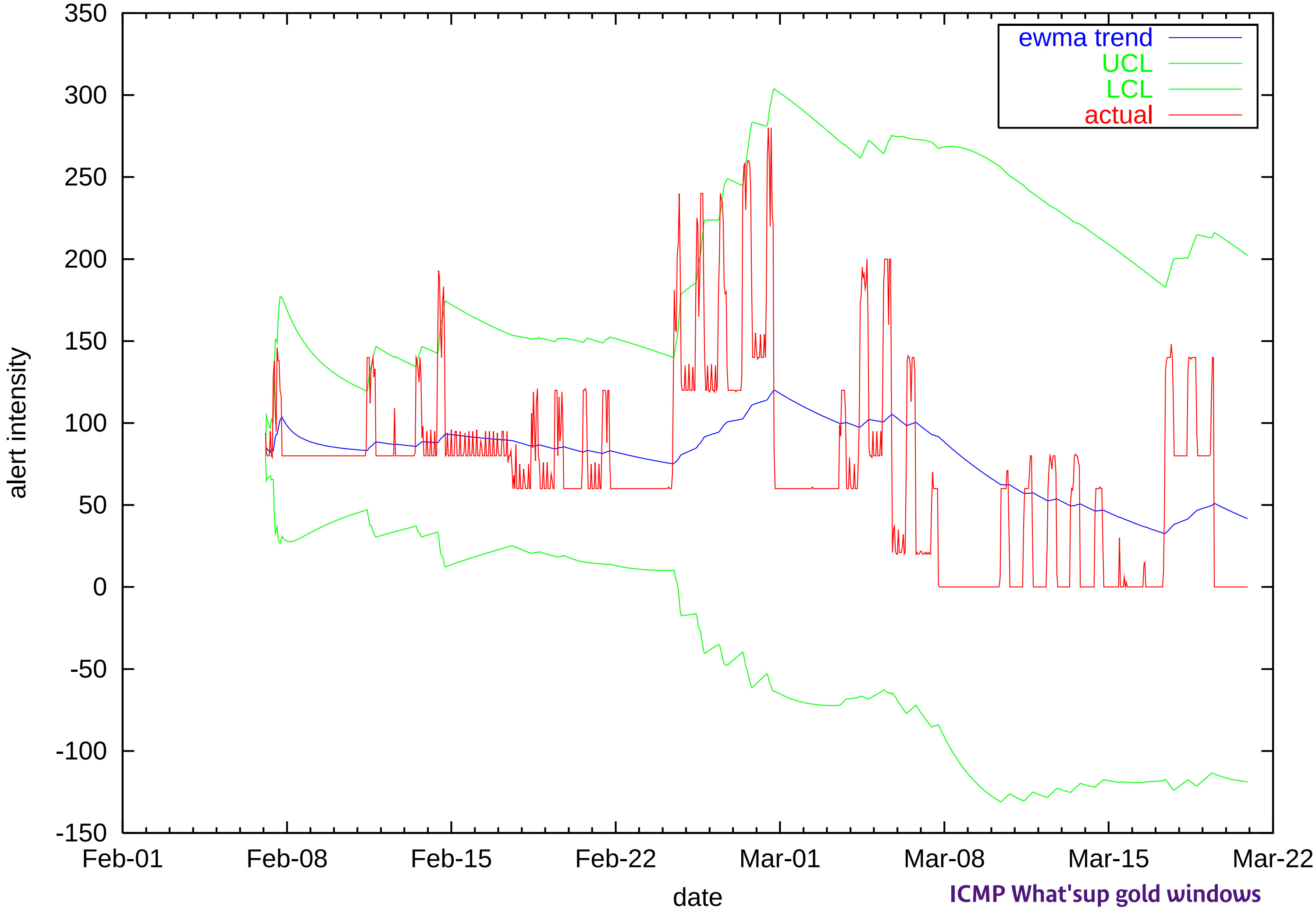
ICMP What'sup gold windows

Local Policy HTTP



Non-explicable alerts

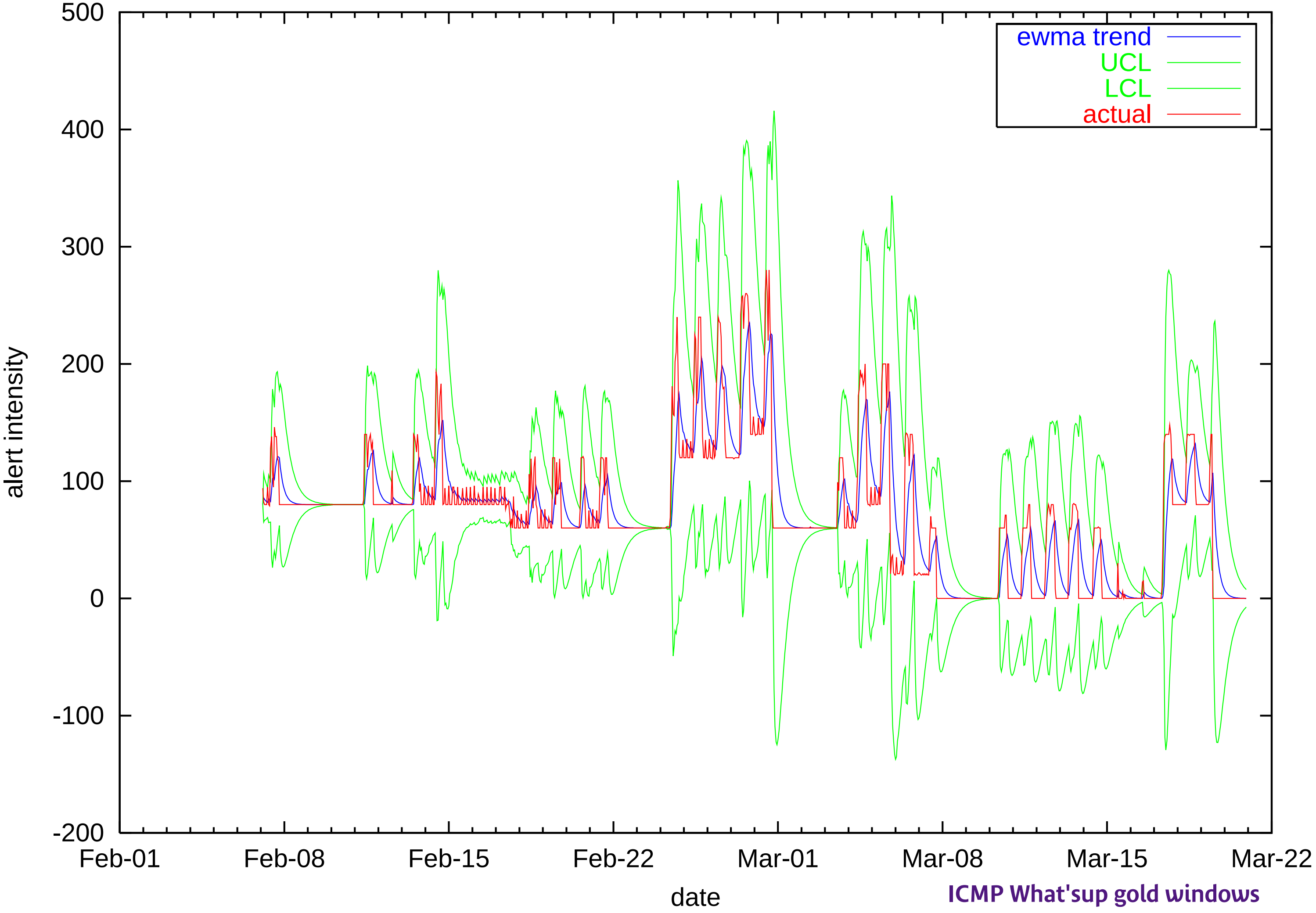
EWMA enveloppe (long)



ICMP What'sup gold windows

(Unrestricted)

EWMA envelopped (short)



ICMP What'sup gold windows

Unrestricted

Distribution of this document is subject to France Telecom's authorization

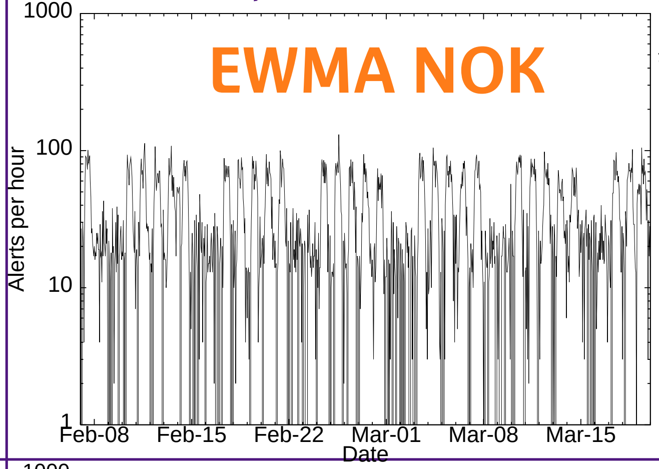
Alert types

Explicable alerts

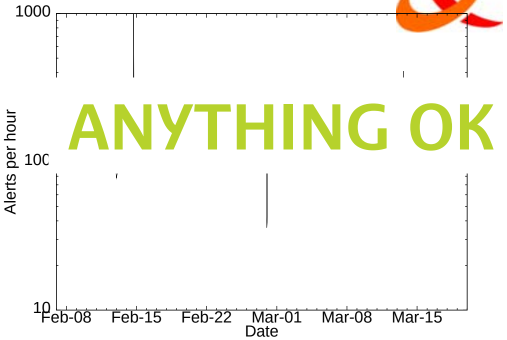
"Well characterized alerts"

- Reference to software vuln
- Low frequency

Speedera (ICMP)



EWMA NOK



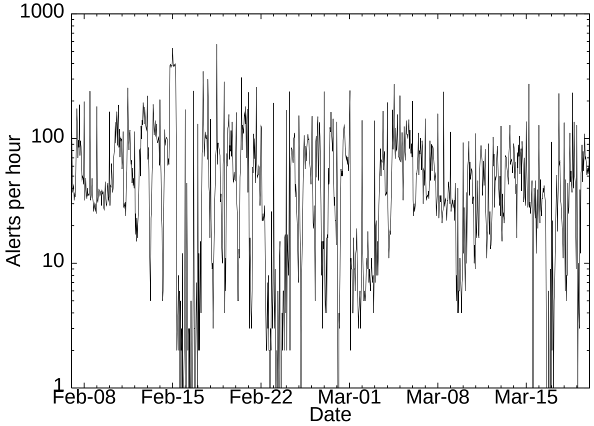
ANYTHING OK

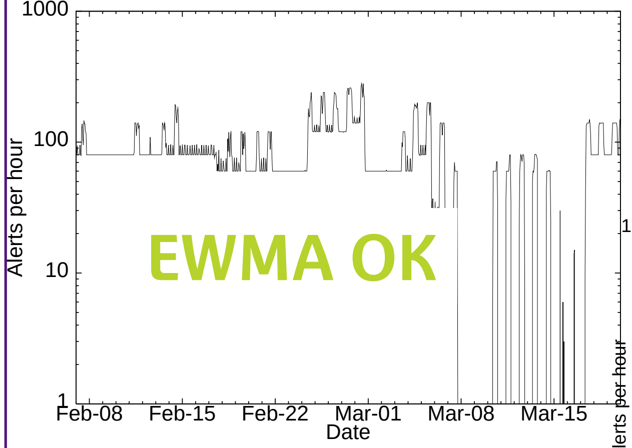
SNMP Request UDP

Low regularity

High regularity

ICMP Destination unreachable
(communication administratively prohibited)

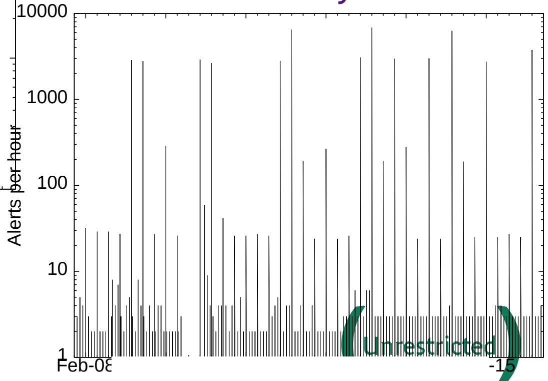




EWMA OK

ICMP What'sup gold windows

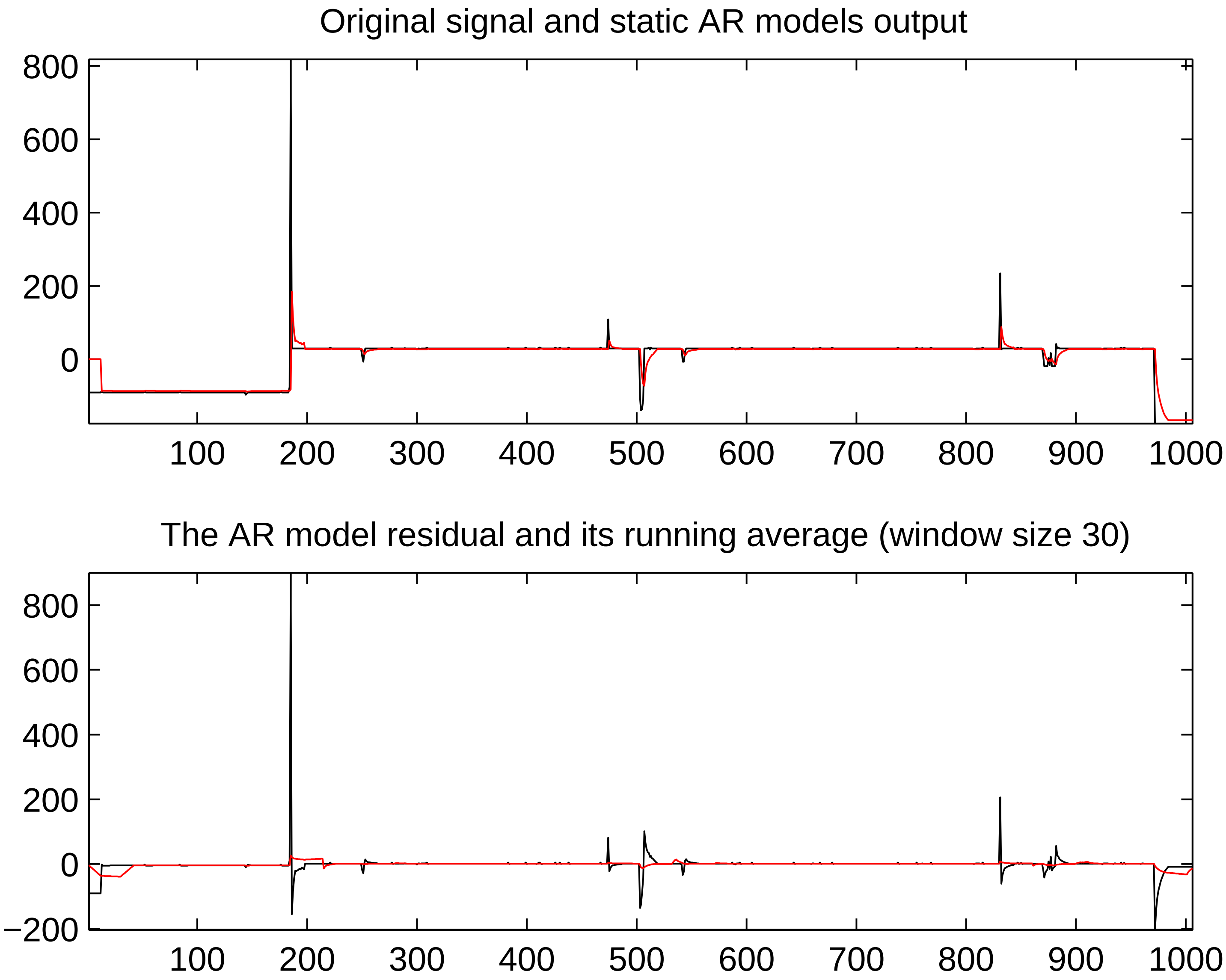
Local Policy HTTP



EWMA NOK

Non-explicable alerts

SNMP AR and kalman filter models

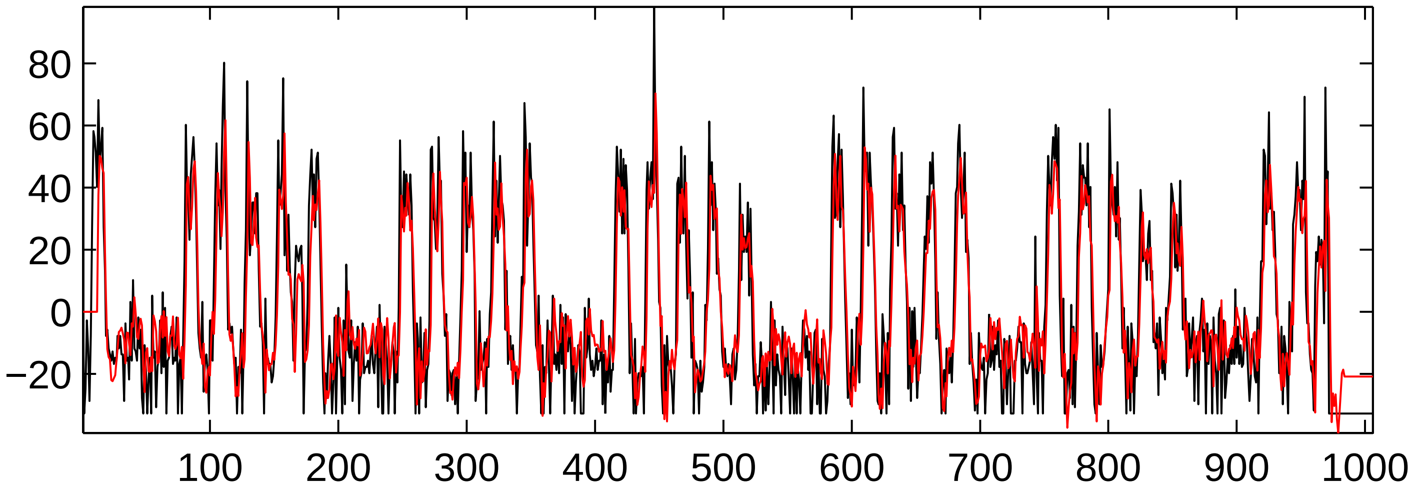


Original signal and static AR models output

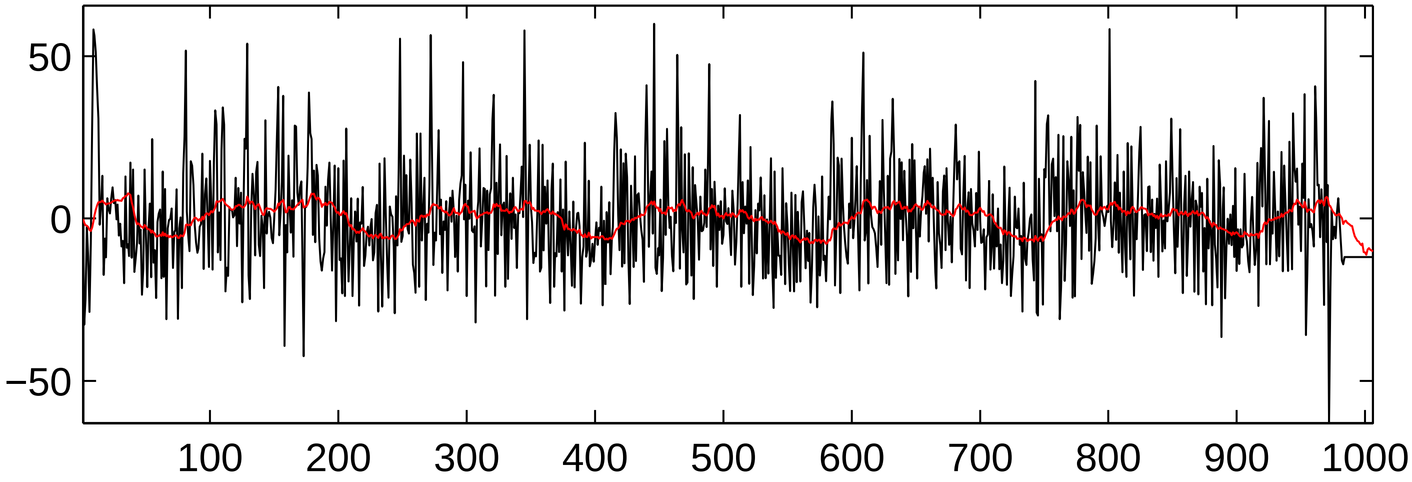
The AR model residual and its running average (window size 30)

Speedera AR and kalman filter models

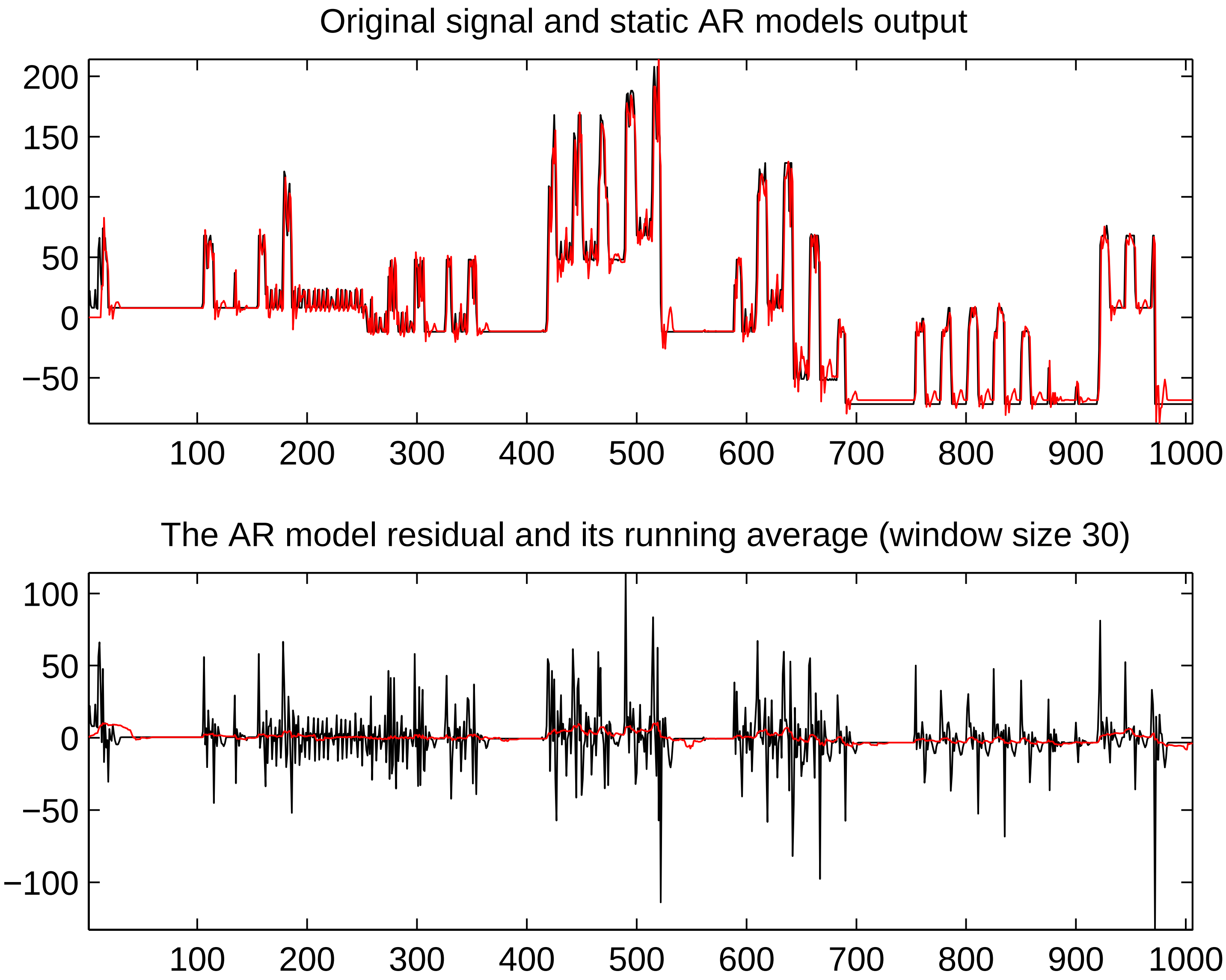
Original signal and static AR models output



The AR model residual and its running average (window size 30)



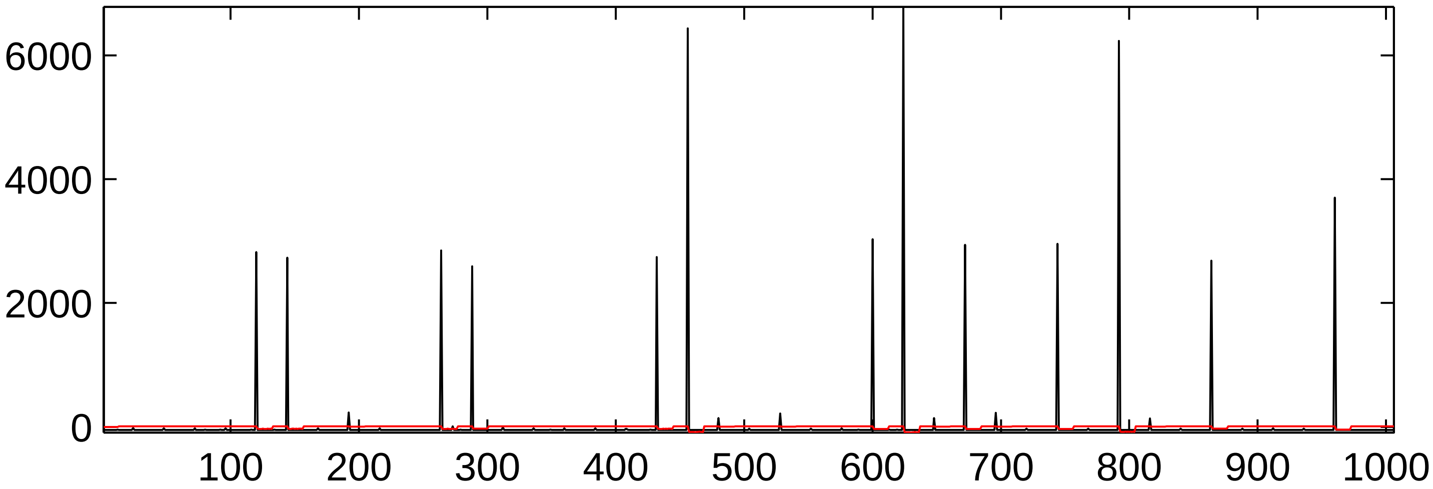
What'sup AR and kalman filter models


Original signal and static AR models output

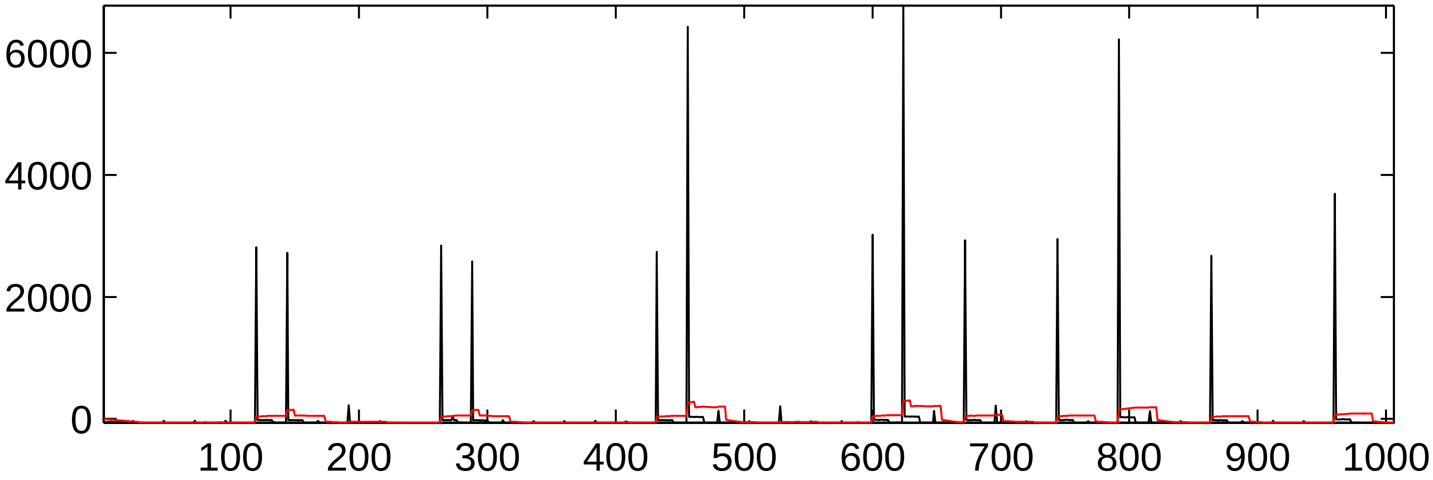
The AR model residual and its running average (window size 30)

Policy AR and kalman filter models

Original signal and static AR models output

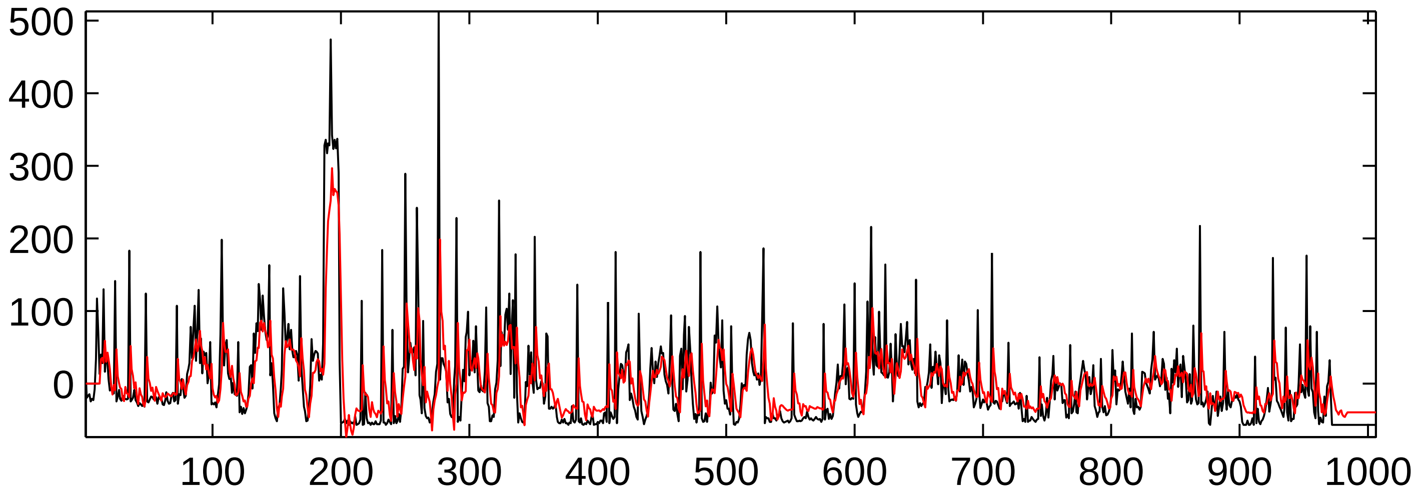


The AR model residual and its running average (window size 30)

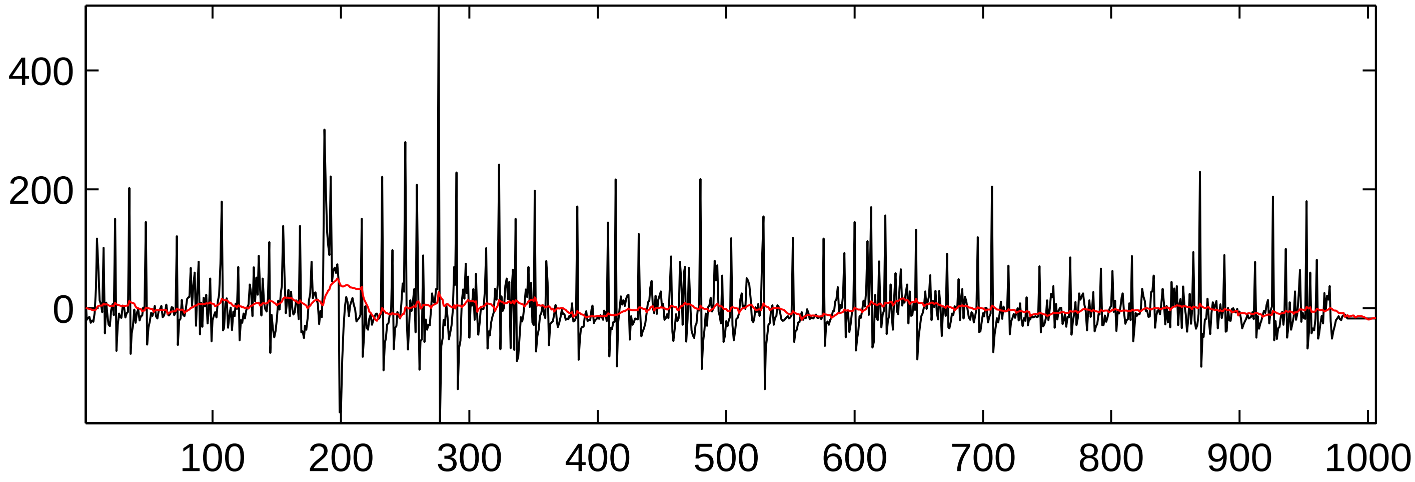


ICMP AR and kalman filter models

Original signal and static AR models output



The AR model residual and its running average (window size 30)



Results

- ⊕ **Reduce "busy intervals" to between 2 and 5%**
- ⊖ **Diagnostic available at the end of the interval**

- ▶ **Future work**
  - Enhanced signal processing techniques for fuzzy behaviour
  - Automatic detection of periodicity
  - Faster messaging to operators
- ▶ **Question: How to qualify abnormal intervals**
  - Source, destination, classification, volume

New challenges for intrusion prevention

New application firewall appliances

- NAI Intrushield
- TippingPoint UnityOne
- ISS Proventia (?)
- See http://www.nss.co.uk/

Requirements

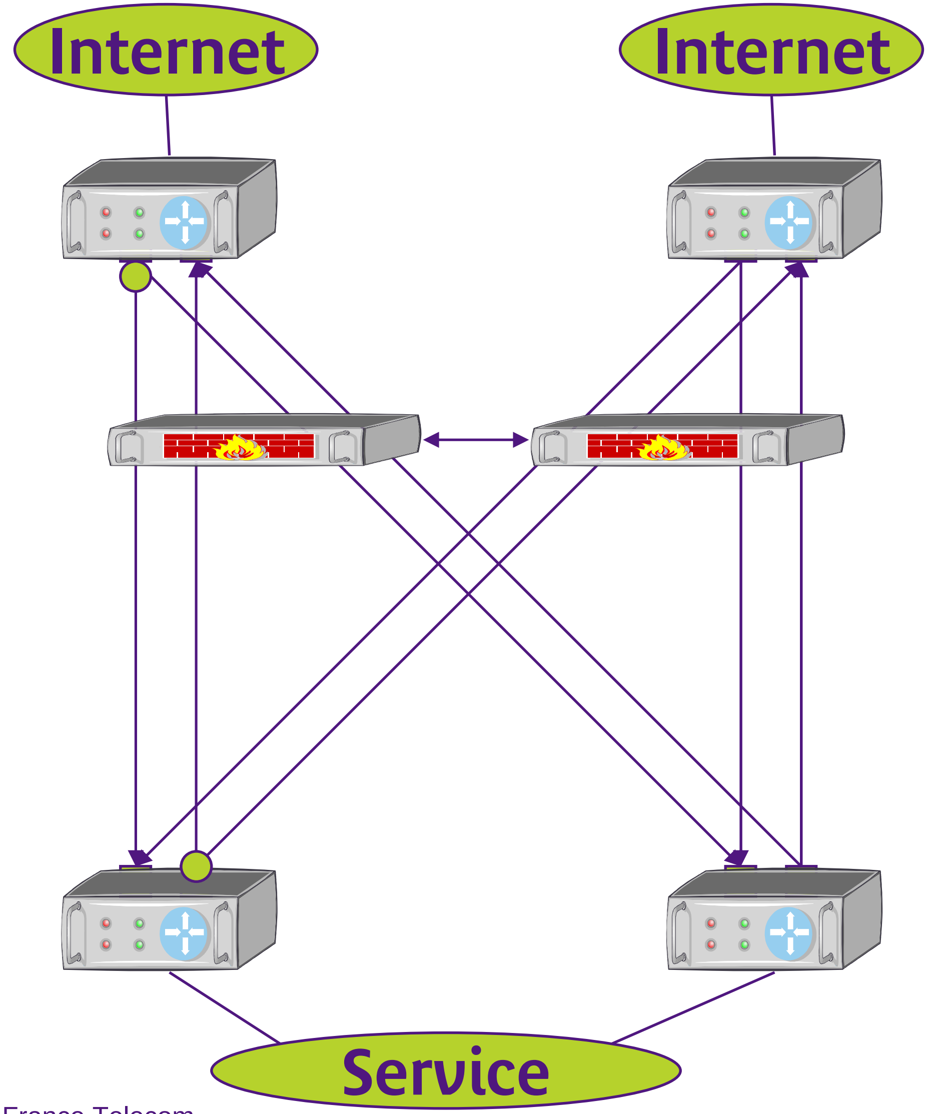
- Capture and analyze 2 1GBps fiber, full duplex
- Complex protocols (entry point of datacenter)
- Integration within the alert management environment

Experimental feeling (Intrushield)

- Nice box, fairly easy to manage
- Can fail with some specially crafted trafic
- Knowledge of protocols exists but limited (performance issues)

(Unrestricted)

Deployment scenario



- Monitor dual homed 10GigE
- Minimum time budget
  - 64 bytes packet
  - 32 nano-seconds
- Parallel activities
  - Packet retrieval
  - Retransmission
  - Context retrieval
  - Analysis
- Challenges:
  - Context structure
  - Analysis algorithm

(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

N(I|P)DS Products (examples)

Established commercial:

- ISS RealSecure
- Cisco CSIDS
- Symantec Manhunt
- NFR NID
- Enterasys Dragon

Open Source:

- Snort
- Prelude NIDS
- Firestorm
- Bro

High speed appliances

- Tipping Point
- Intruvert/McAfee Intrushield
- Netscreen

DDoS dedicated

- Riverhead
- Arbor

(Unrestricted)

H(I|P)DS Products (examples)

Commercial:

- ISS RealSecure Server Sensor
- Cisco Entercept
- Enterasys Dragon Host Sensor
- NFR HID
- OKENA StormWatch

Open source:

- ASAX
- IDIOT
- LIDS
- Prelude LML

(Unrestricted)

Distribution of this document is subject to France Telecom's authorization

Alert management products

Commercial

- Kane Secure Enterprise
- IBM Tivoli Risk Manager
- ISS Site Protector
- NetIQ Security Manager
- Guarded Net neuSecure
- NetForensics
- Intellitactics Network security manager
- Enterasys Dragon Squire
- e-Security e-Sentinel
- …

Open source:

- Prelude manager
- BASE (ex ACID)
- OSSIM
- Threatman
- Anvaal
- Sguil

French

- SOCBox (iv2)
- EAS (Exaprotect)
- Netsecure Log
- …

INTEGRATION REQUIRED

(Unrestricted)

Conclusion

Technologies coming to the real world

- Near real time monitoring
- Concentration of multiple sources of information
- Industrial strenght platforms available
- Open-source solutions available as alternatives

Emergence of SIM consoles

- Nomenclature of attacks and alert names
- Classification of attack types and results
- Correlation is mostly an event-based programming engine

Missing technologies

- Simple, secure logging mechanisms
- Simulation of security incidents
- Forecast of security incidents (pro-active)

(Unrestricted)

Perspectives

- **Integrate better the security policy with monitoring techniques**
  - Increase the number of control points (PEP)
  - Interconnect counter-measures and security policy
  - Protect the information, not only the infrastructure
- **Two main families of attack processes**
  - Automated (viruses, worms)
    - Automated countermeasures the only way
    - Compromise between availability and confidentiality (integrity)
  - Manual attack processes "under the radar"
    - Improve our detection capability
    - Take into account legal constraints
- **Go beyond hard shells and perimeter security**
  - Improve dialog between security components

(Unrestricted)