Intrusion Detection Systems Correlation: a Weapon of Mass Investigation

Pierre Chifflier Sébastien Tricaud

INL
101/103 Bvd MacDonald
75019 Paris, France

CanSecWest Vancouver 2008

Pierre Chifflier, Sébastien Tricaud

INL - 101/103 Bvd MacDonald - 75019 Paris, France

What are IDSs?

- Intrusion Detection Systems
- Marketing folks may call it
  - Intrusion Prevention System (IPS)[1]
  - Security Information and Event Management (SIEM)
- Since IPS and SIEM sound too 2005, we stick to IDS

[1] To prevent an attack, we should first detect it ;)

What are they?

- Host IDS (HIDS): Not (really) prone to false positives
- Network IDS (NIDS): Cannot decrypt unknown encrypted traffic, is **not** the target machine and sensitive to false positives
- Hybrid IDS (HbIDS): Mixes HIDS and NIDS

Interesting sources of information out there

Why do we keep our interest in **Hybrid IDS** when we have **more than just NIDS and HIDS** ?

Interesting sources of information out there

Why do we keep our interest in **Hybrid IDS** when we have **more than just NIDS and HIDS** ?

Low Level Sources:

- **Routers**: Cisco, Linksys, Juniper, ...
- **Firewalls**: Netfilter, NuFW, Checkpoint, pf, ...
- **Operating systems**: System logs, users, running applications, ...
- **Physical**: Alarm, ...

Interesting sources of information out there

Why do we keep our interest in **Hybrid IDS** when we have **more than just NIDS and HIDS** ?

Low Level Sources:

- **Routers**: Cisco, Linksys, Juniper, ...
- **Firewalls**: Netfilter, NuFW, Checkpoint, pf, ...
- **Operating systems**: System logs, users, running applications, ...
- **Physical**: Alarm, ...

High Level Sources:

- **Honeypots**: Nepenthes, ...
- **Network**: Snort, Sancp, NuFW, ...
- **Host**: Auditd (SELinux), Linux PAM, Samhain, Ossec, Prelude LML, ClamAV ...
- **Scanners**: Nessus, p0f, nmap ...

Meta IDS (MIDS)

Hybrid IDS

An Hybrid IDS combines HIDS and NIDS.

Meta IDS

A Meta IDS (MIDS) mixes any element that can send data useful for intrusion detection as a whole

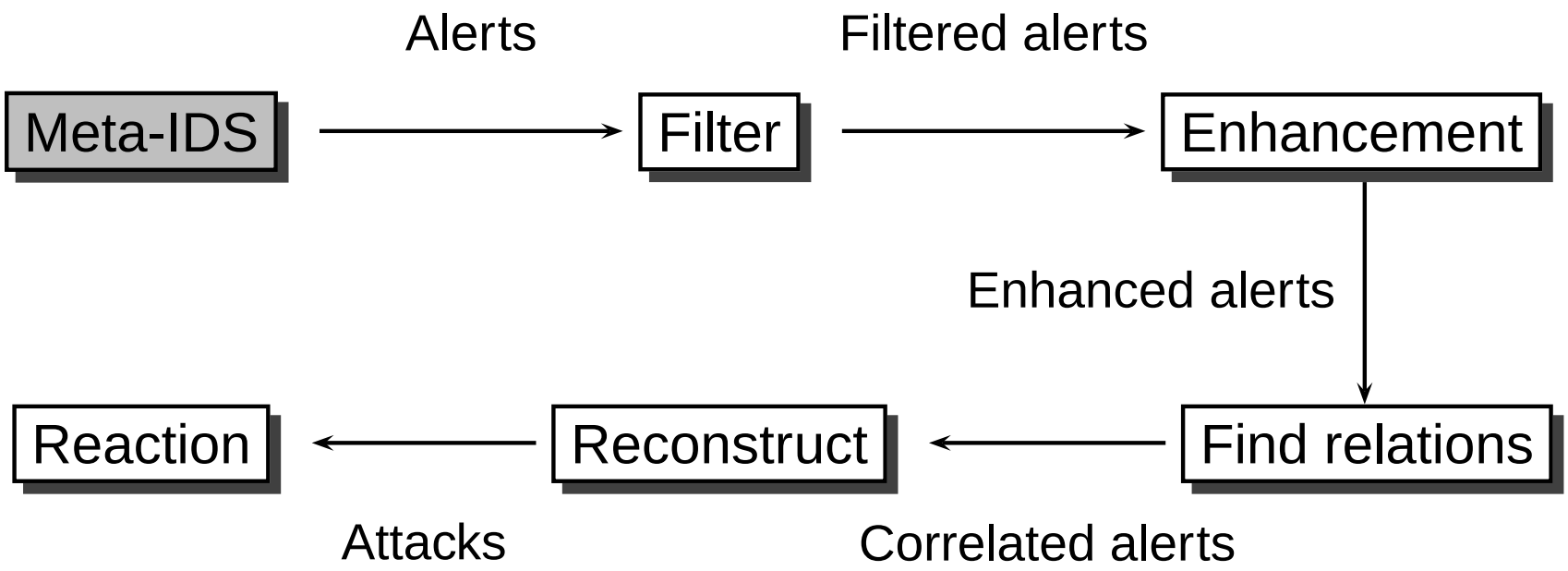
Prelude IDS

Prelude IDS has evolved to a Meta IDS

Examples of alerts :

- OSSEC: `SSHD authentication success.`
- Prelude LML: `Admin login successful`
- Snort: `BLEEDING-EDGE SCAN NMAP -f -sS`
- ClamAV: `Eicar-Test-Signature (succeeded)`
- Auditd (SE Linux): `App Abnormal Termination`

Correlation path



What everybody knows: IDS limitations

- Too much information
- Limited view
- False positives
- False negatives
- Evasion (fragmentation, signature, time, . . .)

IDS correlation

- To limit IDS pitfalls, we need correlation
  - We need a Meta-IDS
  - We need a scalable and distributed architecture to centralize information
  - We need to define accurately each alert and each agent

The IDMEF: Intrusion Detection Message Exchange Format

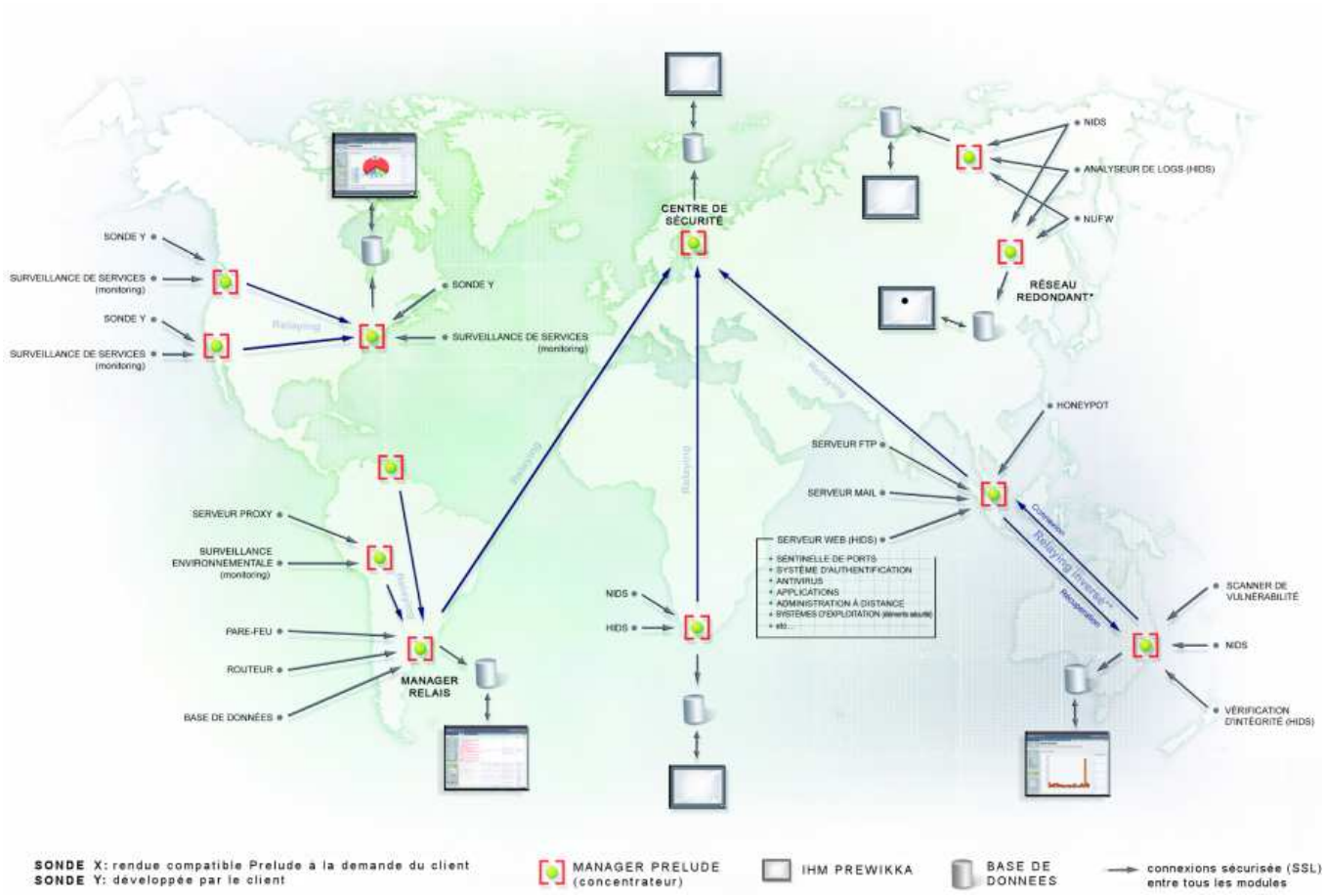
- Normalize agent alerts regardless of their nature
  - Alert information is inherently heterogeneous
  - Intrusion detection environments are different
  - Analyzer capabilities are different
  - Operating environments are different
  - Commercial vendor objectives are different
- Provides an exhaustive vocabulary to IDS developers and users

$\Rightarrow$ IDMEF (RFC 4765)
`http://www.rfc-editor.org/rfc/rfc4765.txt`

Prelude IDS

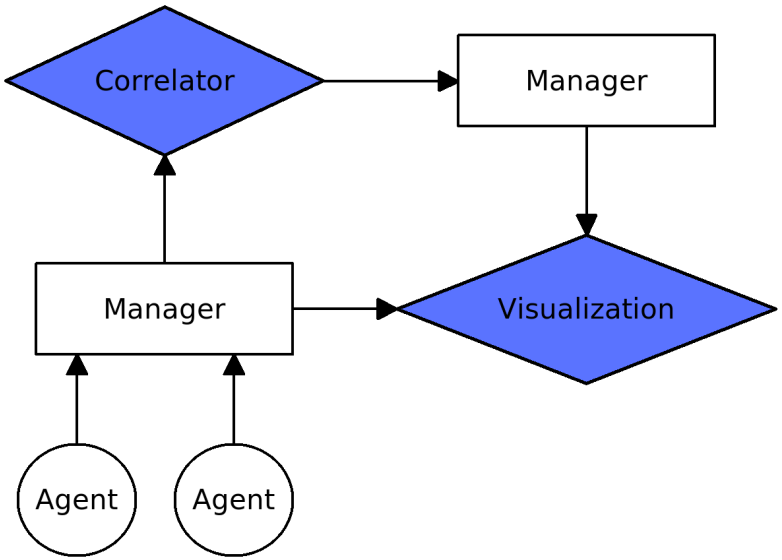
- Meta-IDS implementing IDMEF
  - libprelude, libpreludedb
  - Prelude LML: Analyze logs
  - Prelude Correlator: Correlate alerts from agents
  - Prelude Manager: Centralize and store/deliver/relay alerts
  - Prewikka: Graphical interface
- Required capabilities for correlation:
  - **Encryption** between agents and manager, manager to manager
  - **Failover**, whenever alerts cannot be sent to the manager
  - **Relaying** to centralize, backup and filter alerts
  - **Reverse relaying** to keep DMZ secure
  - **Normalize** your alerts: Complete the IDMEF message

The correlation challenge



Pierre Chifflier, Sébastien Tricaud
INL - 101/103 Bvd MacDonald 75019 Paris, France

Prelude user architecture



Pierre Chifflier, Sébastien Tricaud

INL - 101/103 Bvd MacDonald - 75019 Paris, France

Objectives

What ?

- Concentrate on high-level analysis
- Reduce noise created by false positives or harmless events
- Fight evasion
- Discover new attacks

Objectives

What ?

- Concentrate on high-level analysis
- Reduce noise created by false positives or harmless events
- Fight evasion
- Discover new attacks

How ?

- Use trust score to improve the reliability
- Combine elements from heterogeneous sources (use the **Meta**-IDS !)
- Reconstruct and understand the attack

Trust score (TS)

$$TS = \text{severity of the alert} \times \text{accuracy of the alert}$$

- 0 (false alarm) $< TS <$ 1 (known and verified attack)
- Initial value depending on the alert (analyzer and signature reliability)
- NIDS: high probability of false alerts $\Rightarrow$ low TS
- Will be adjusted during correlation steps
- Will be used to take the final decision

Understand an attack

Objectives :

- Reconstruct the sequence of events
- Detect the targets, protocols, tools, . . .
- Adapt the severity
- Reduce false positives
- Prepare for an eventual counter-measure
- Ensure the Security Policy is properly applied

Understand an attack

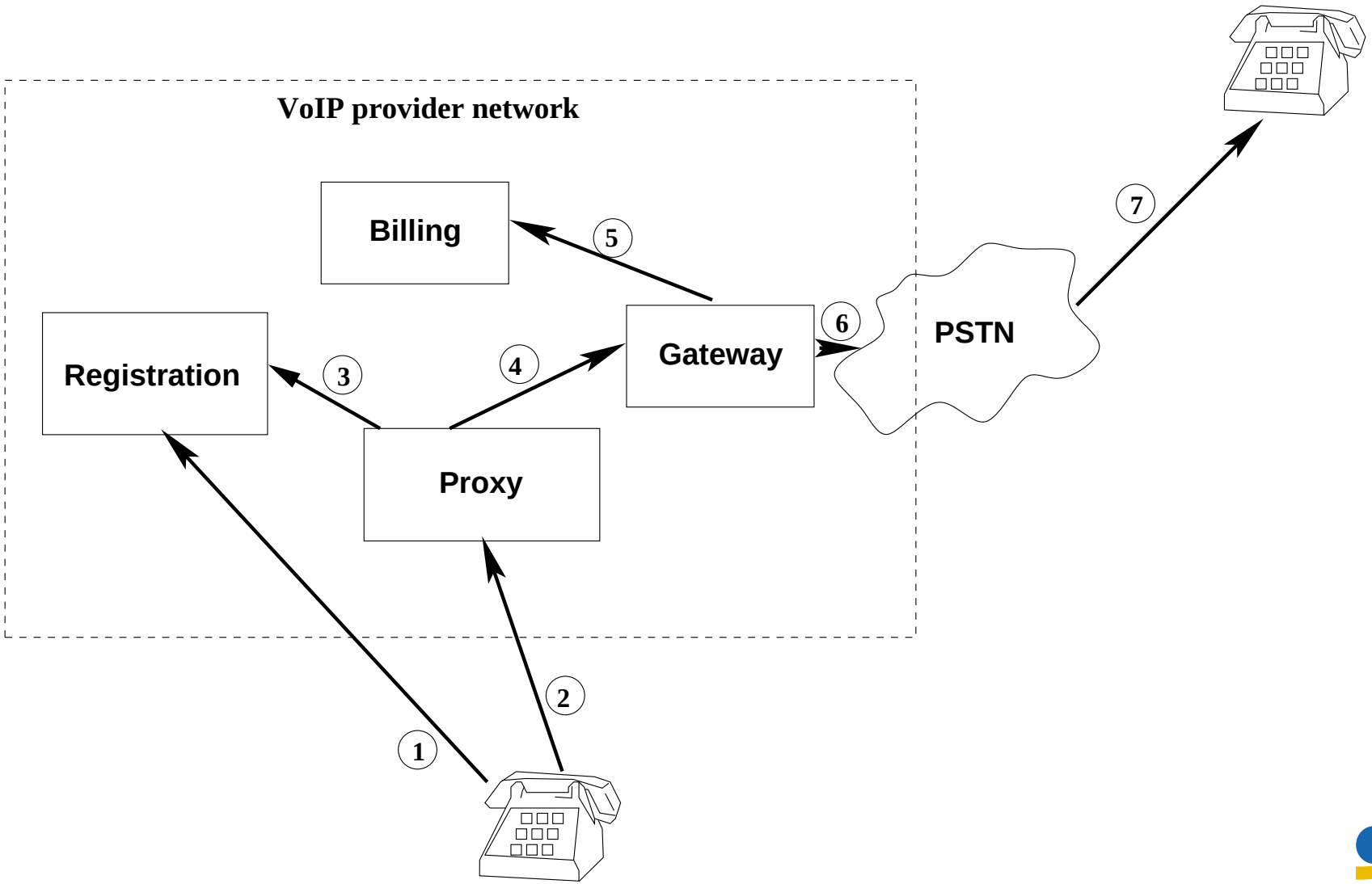
Objectives :

- Reconstruct the sequence of events
- Detect the targets, protocols, tools, ...
- Adapt the severity
- Reduce false positives
- Prepare for an eventual counter-measure
- Ensure the Security Policy is properly applied

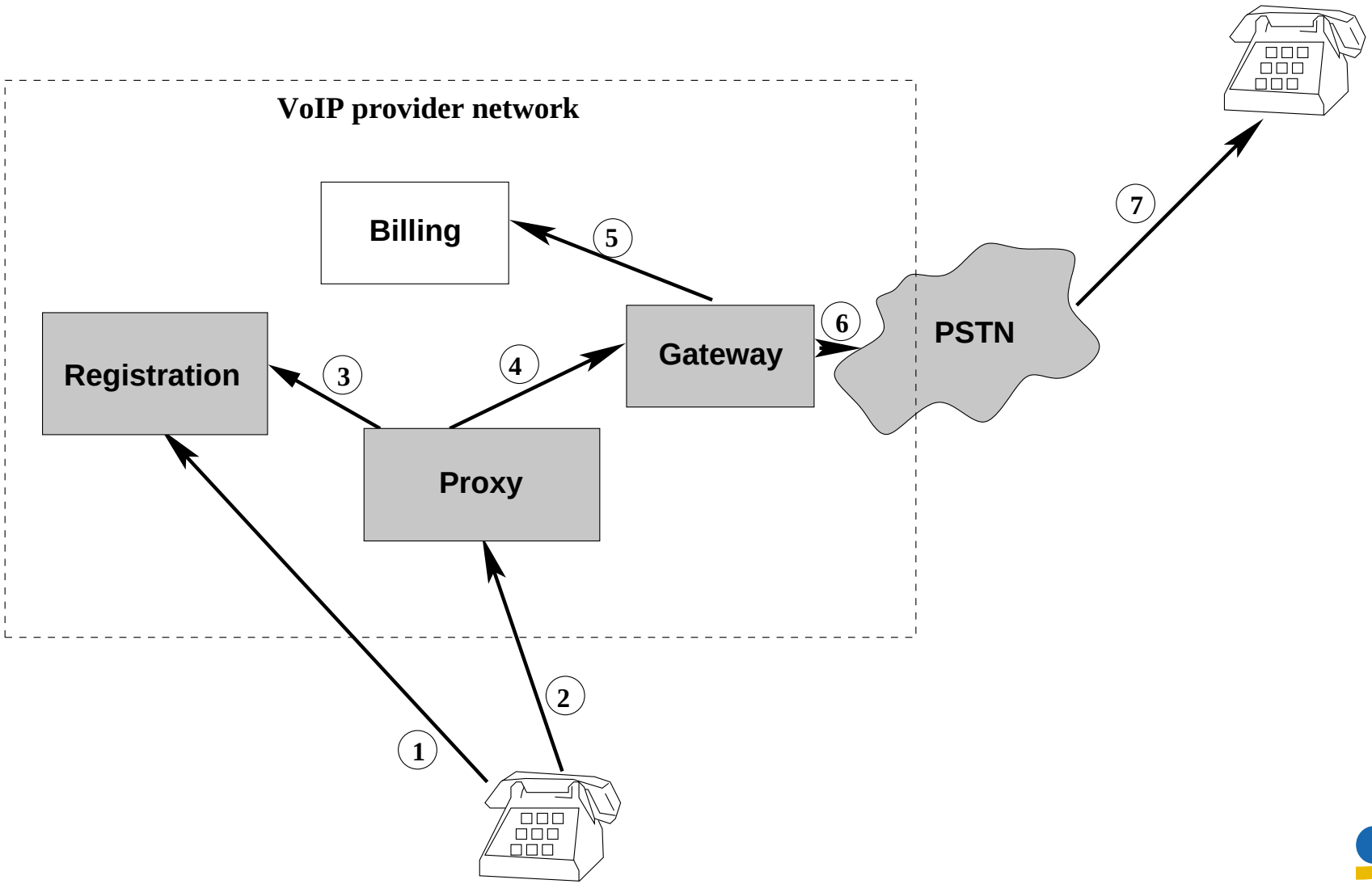
Tools:

- Normalization, Centralization
- **Correlation**
- **Visualization**

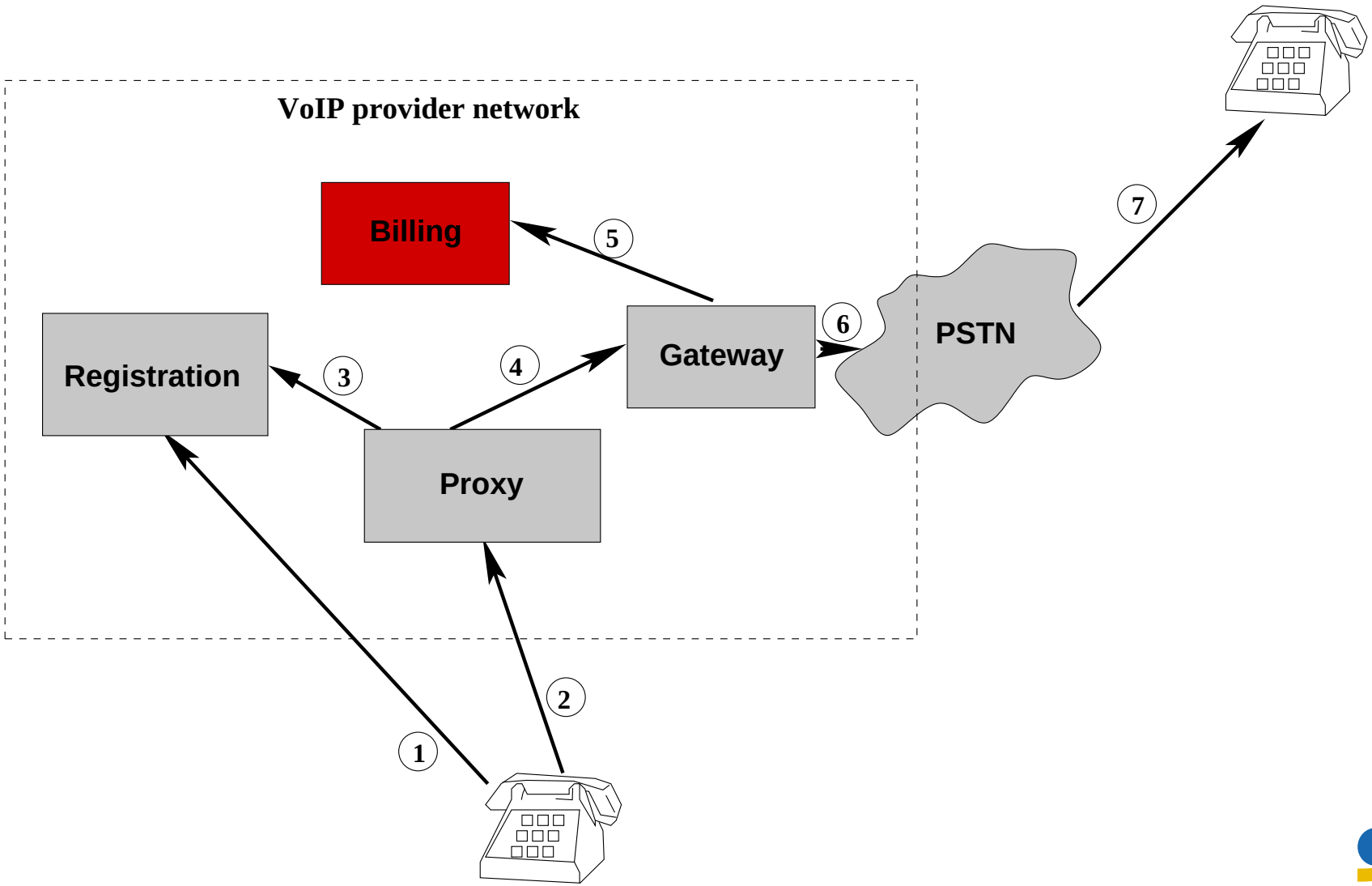
Correlation

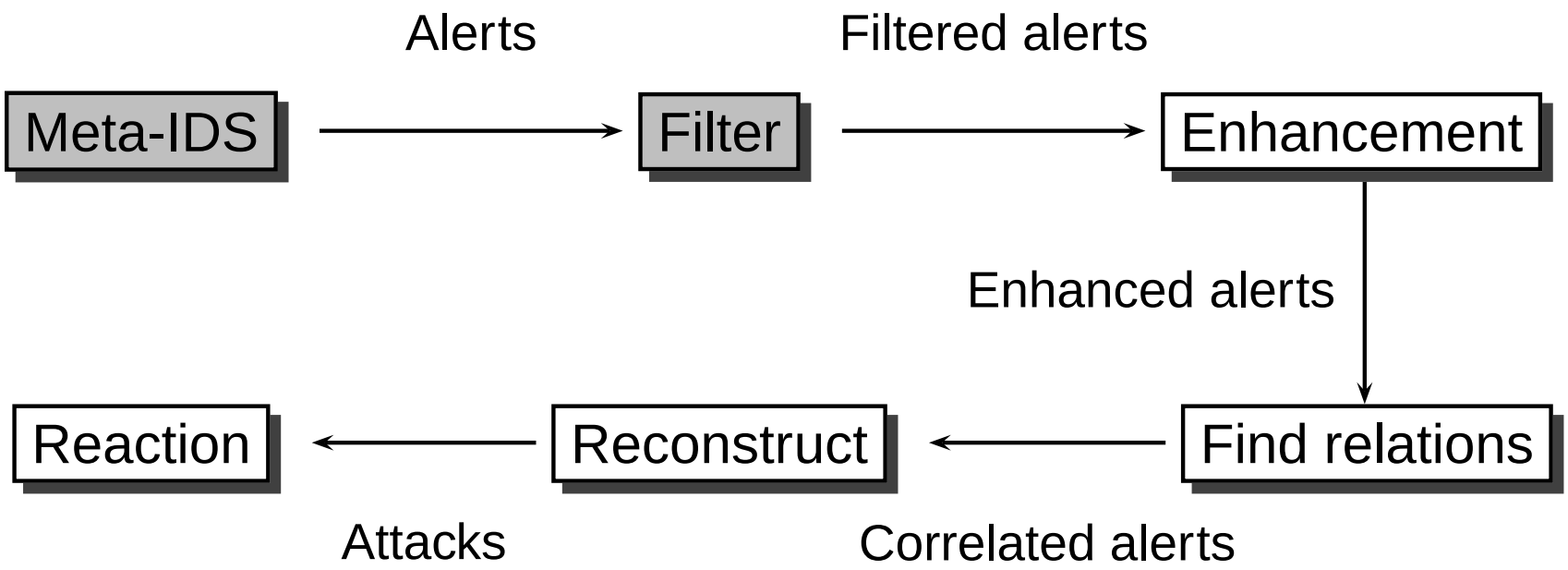


Correlation

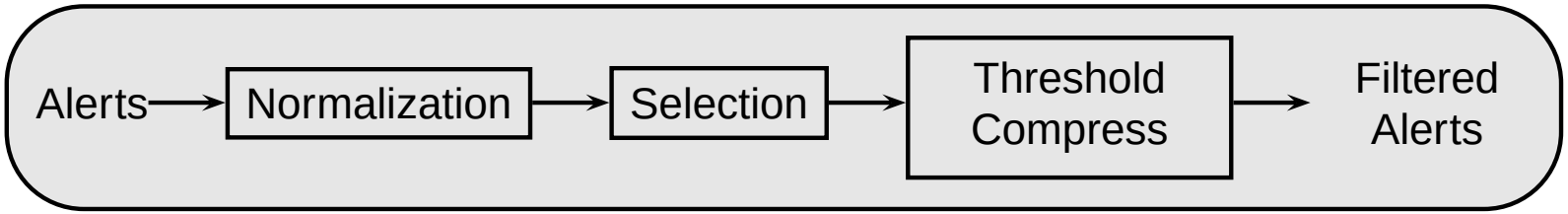


Correlation

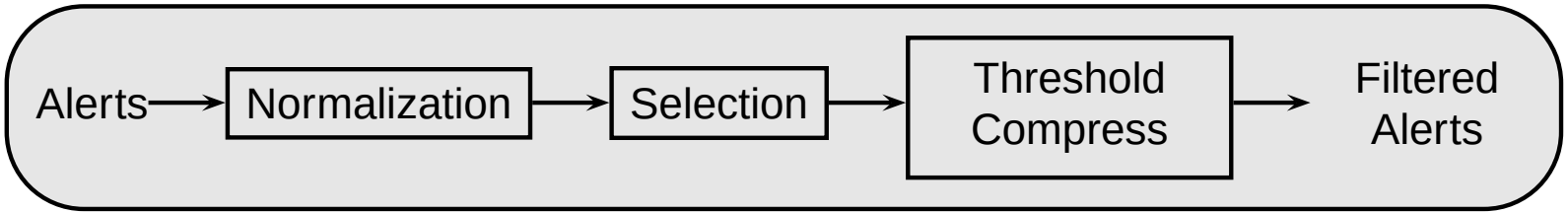



Alerts
Filtered alerts
Meta-IDS
Filter
Enhancement
Enhanced alerts
Reaction
Reconstruct
Find relations
Attacks
Correlated alerts

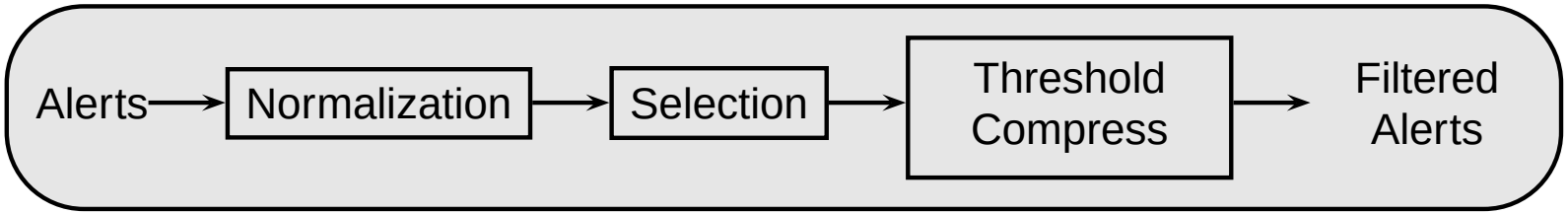
Filtering



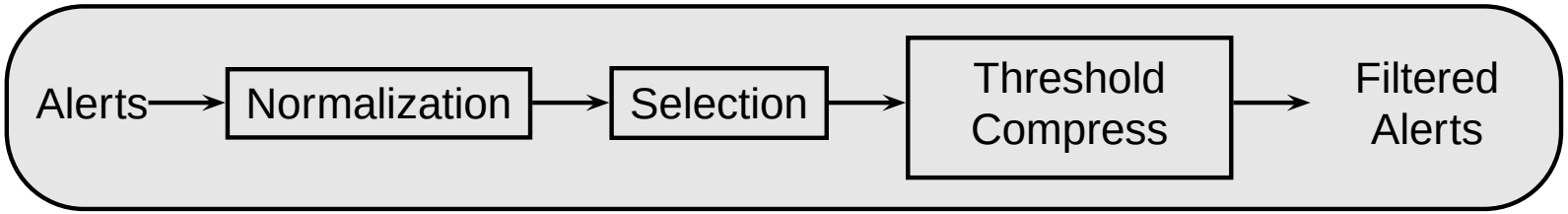
- Normalize input (*classification.text*, *analyzer type*)
- Apply initial filtering
- Compression: replace n alerts by one, keeping all information
- Threshold: if $n > threshold$, ignore other alerts (loosing information)



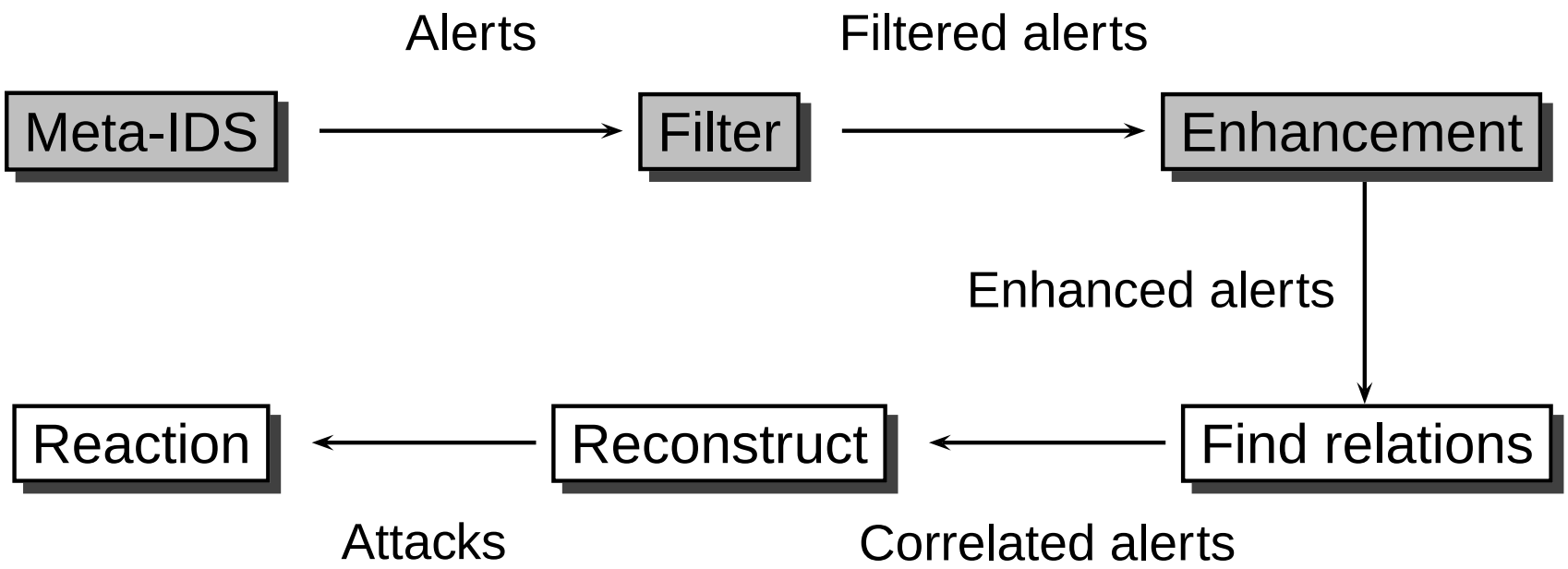
Alert	Filtered alert
SSHD authentication success	User login attempt completion: success



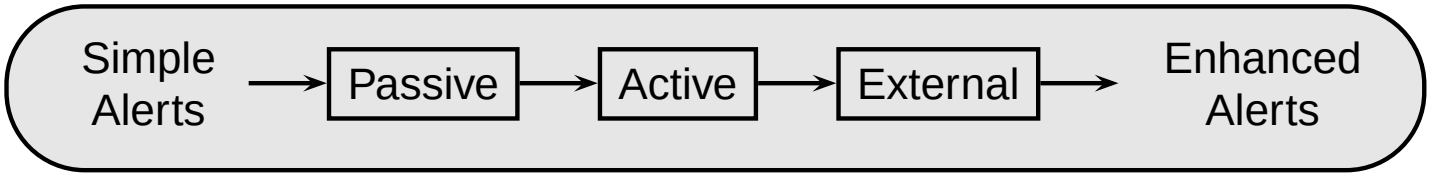
Alert	Filtered alert
SSHD authentication success	User login attempt completion: success
User login failed (Alice) User login failed (Alice)	User login attempt (2 × Alice) completion: failed



Alert	Filtered alert
SSHD authentication success	User login attempt completion: success
User login failed (Alice) User login failed (Alice)	User login attempt (2 × Alice) completion: failed
User login successful (Alice)	*dropped*


Alerts
Filtered alerts
Meta-IDS
Filter
Enhancement
Enhanced alerts
Reaction
Reconstruct
Find relations
Attacks
Correlated alerts

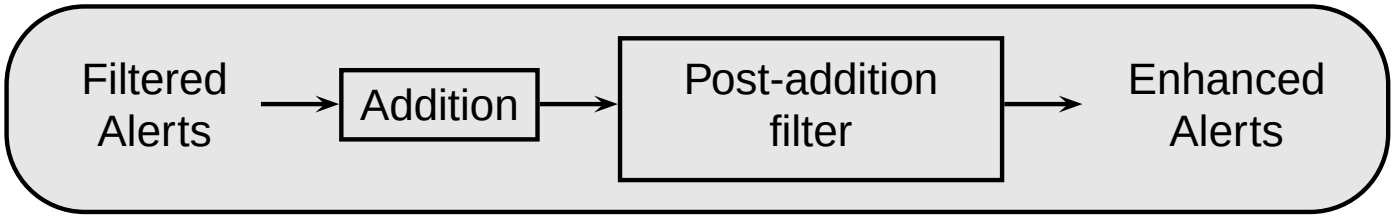
Enhancement (enlarge your alerts)



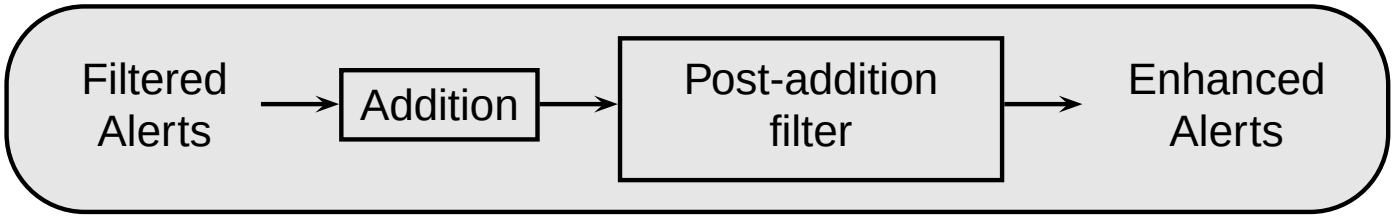
Passive Infomation Collection (PIC):

- Passive data (OS, applications, versions, inventory)
- Profiling (sancp)
- OSVDB, BID, CVE, patches, known exploits
- Current attacks (DShield)
- Passive . . . or not ! (*hint: Nessus*)

Post-enhancement filter



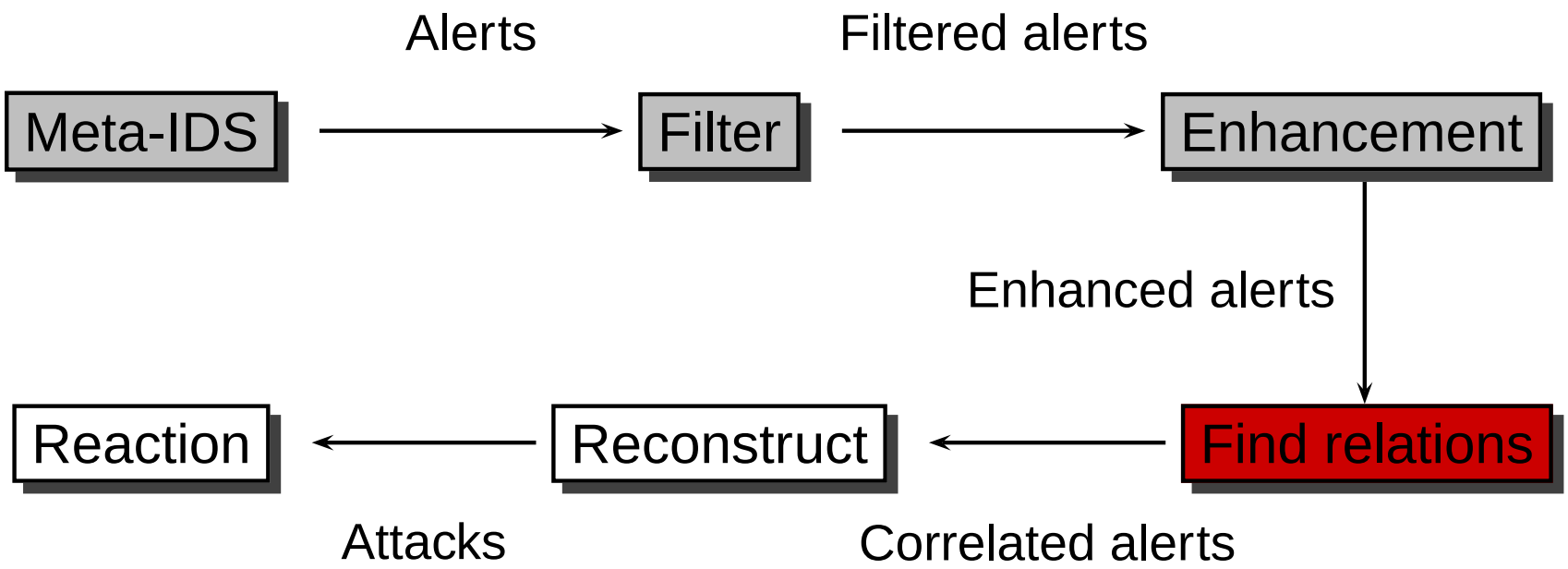
- Send alerts on spurious changes
- Re-evaluate alert with additional data
  - Delete alert or lower trust score if the target is not affected
  - Increase trust score if affected



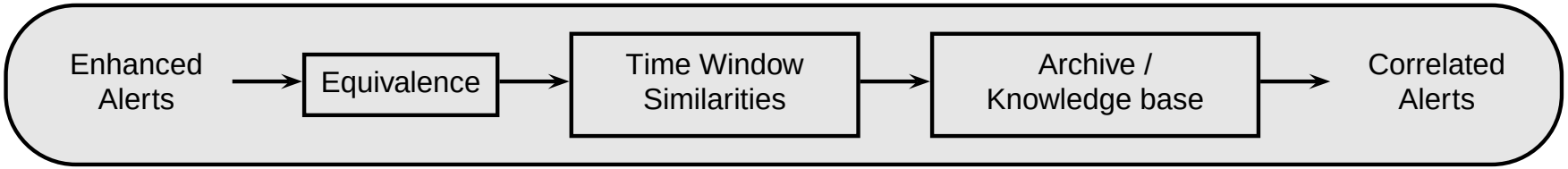
Filtered alert	Enhanced alert
"THCIISLame IIS SSL Exploit Attempt"	"THCIISLame IIS SSL Exploit Attempt" Host OS: Linux 2.6.24 Reference: isc.sans.org/diary.php?date=2004-07-17 Exploit www.thc.org/exploits/THCIISSLame.c *dropped*

Attack definition

- An attack is a sequence of alerts or events with a particular relation
- $Attack = n \times$ alerts
- $n \geq 1$
- Classification of the *attack* can be done *after* the entire correlation

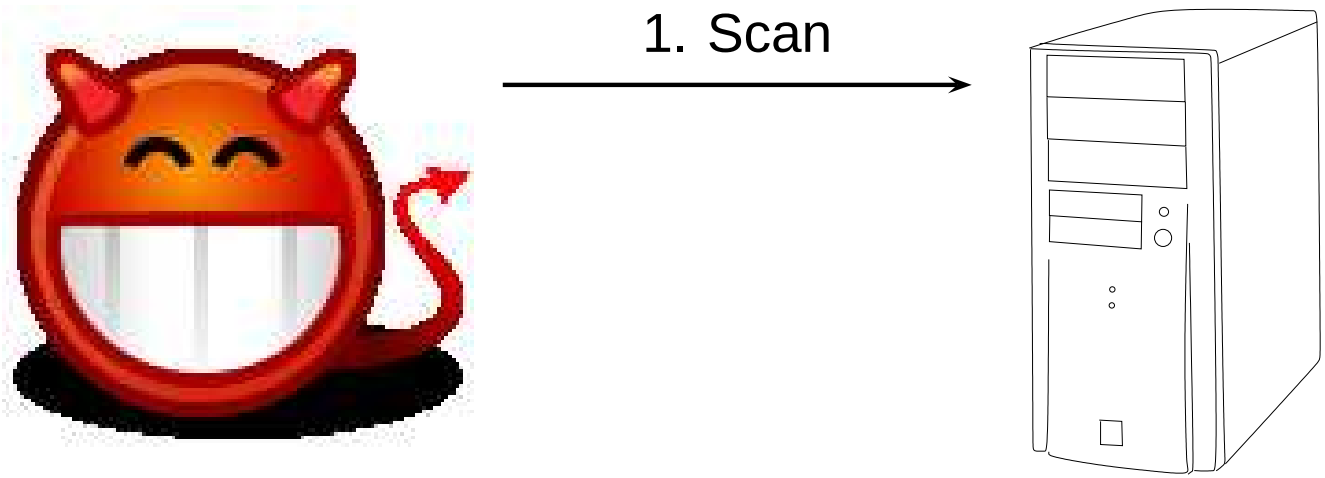

Alerts
Filtered alerts
Meta-IDS
Filter
Enhancement
Enhanced alerts
Reaction
Reconstruct
Find relations
Attacks
Correlated alerts

Find relations

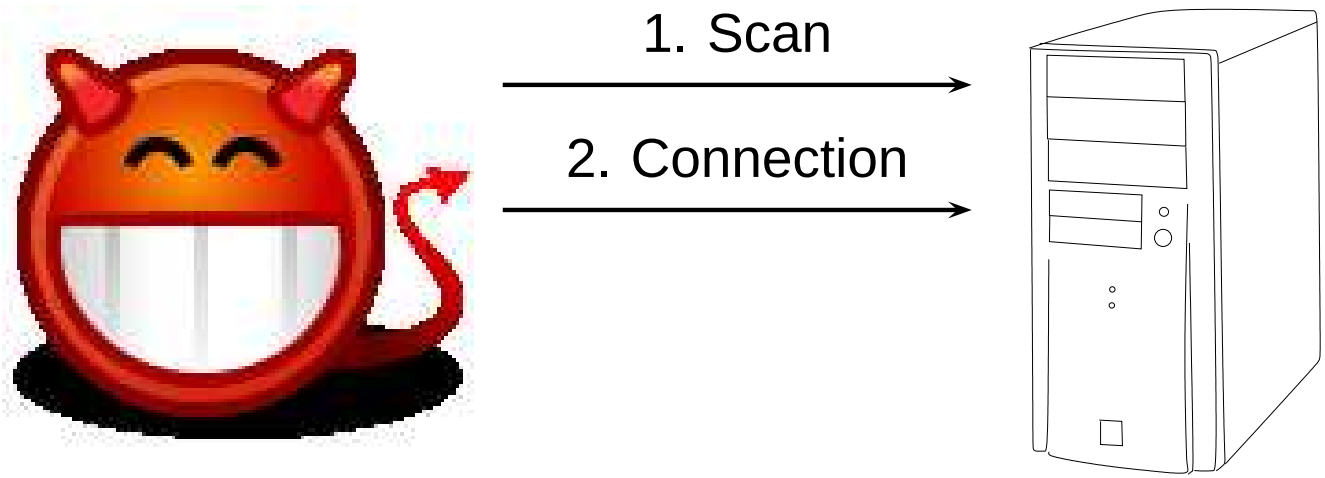


- Equivalence
- Similarities, during a time window (source, destination, attack vector, . . .)
- Archive / knowledge database (known patterns)
- Search on a long time range
- Regular events

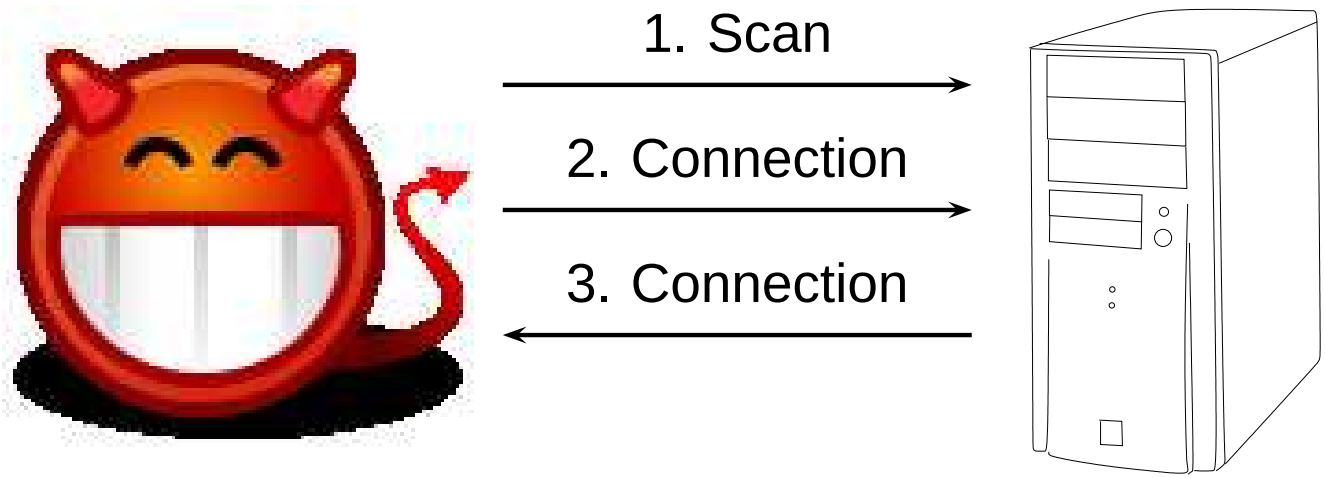
Find relations



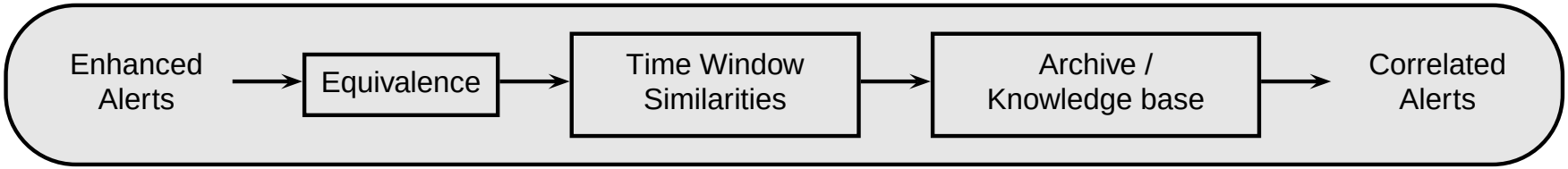
Find relations



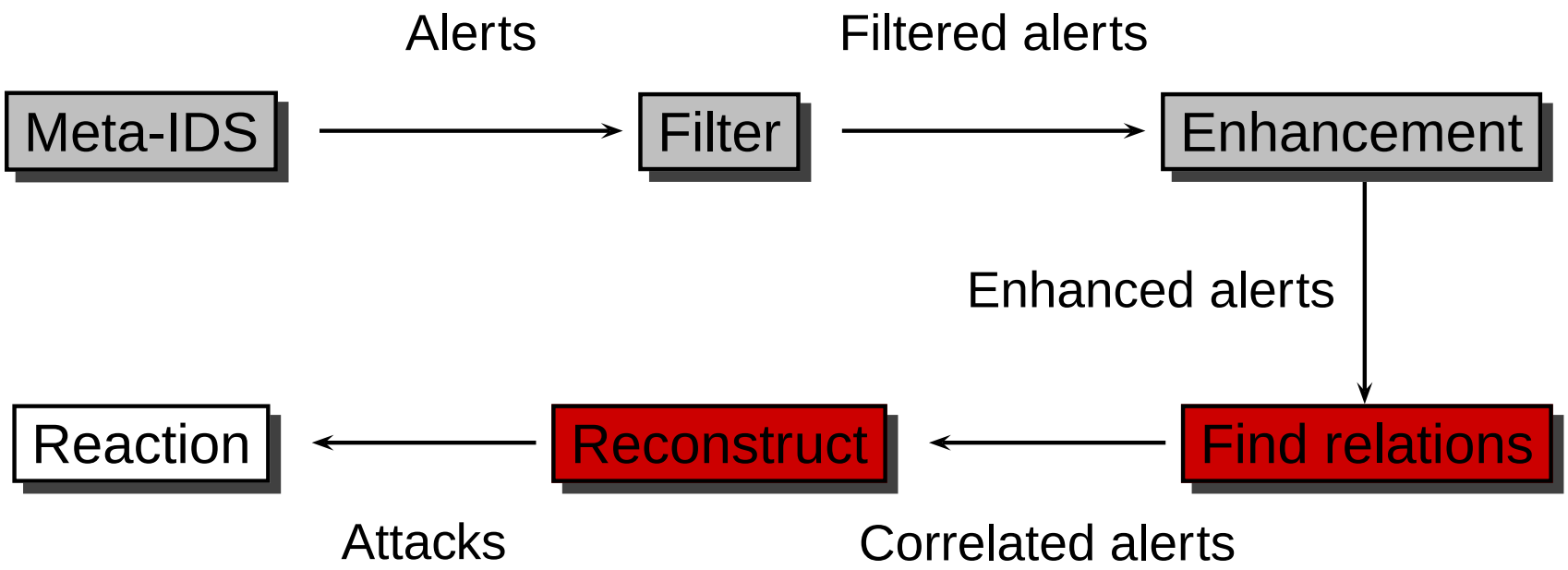
Find relations



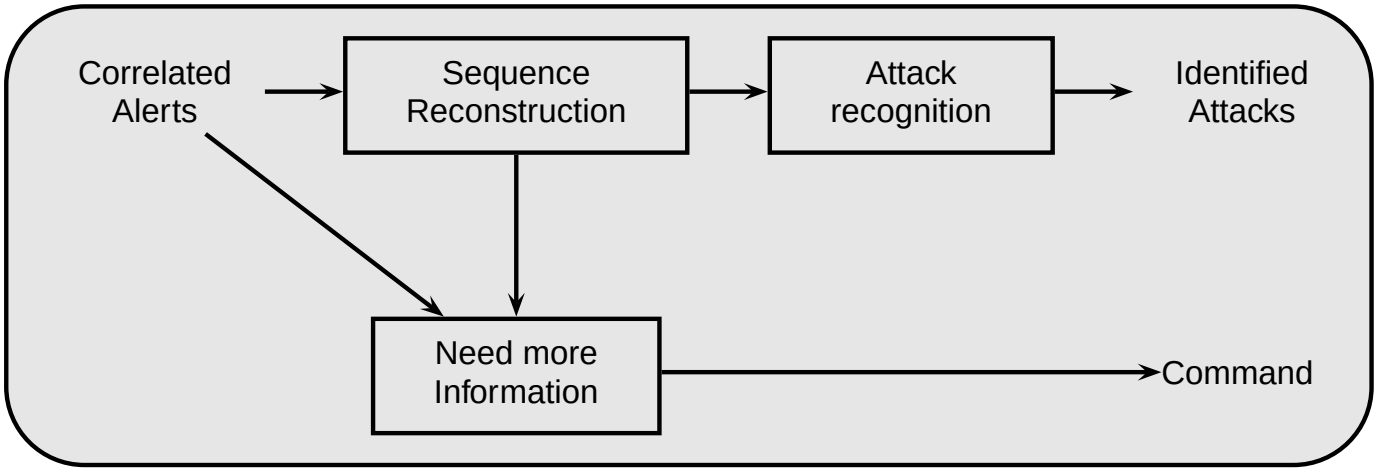
Find relations



Enhanced Alert	Correlated alert
Port scan + Incoming connection + Outgoing connection *source/dest*	Sequence 3 elements
OSSEC SSHD authentication success (Alice) + Prelude LML User login successful (Alice)	SSH login attempts ($1 \times$ Alice)

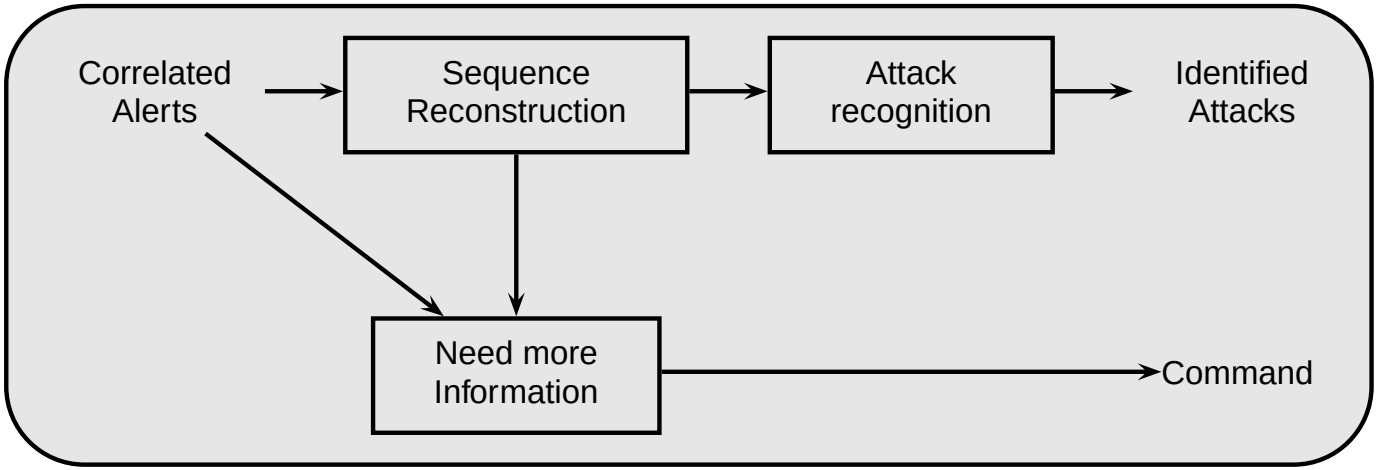

Alerts
Filtered alerts
Meta-IDS
Filter
Enhancement
Enhanced alerts
Reaction
Reconstruct
Find relations
Attacks
Correlated alerts

Attack reconstruction

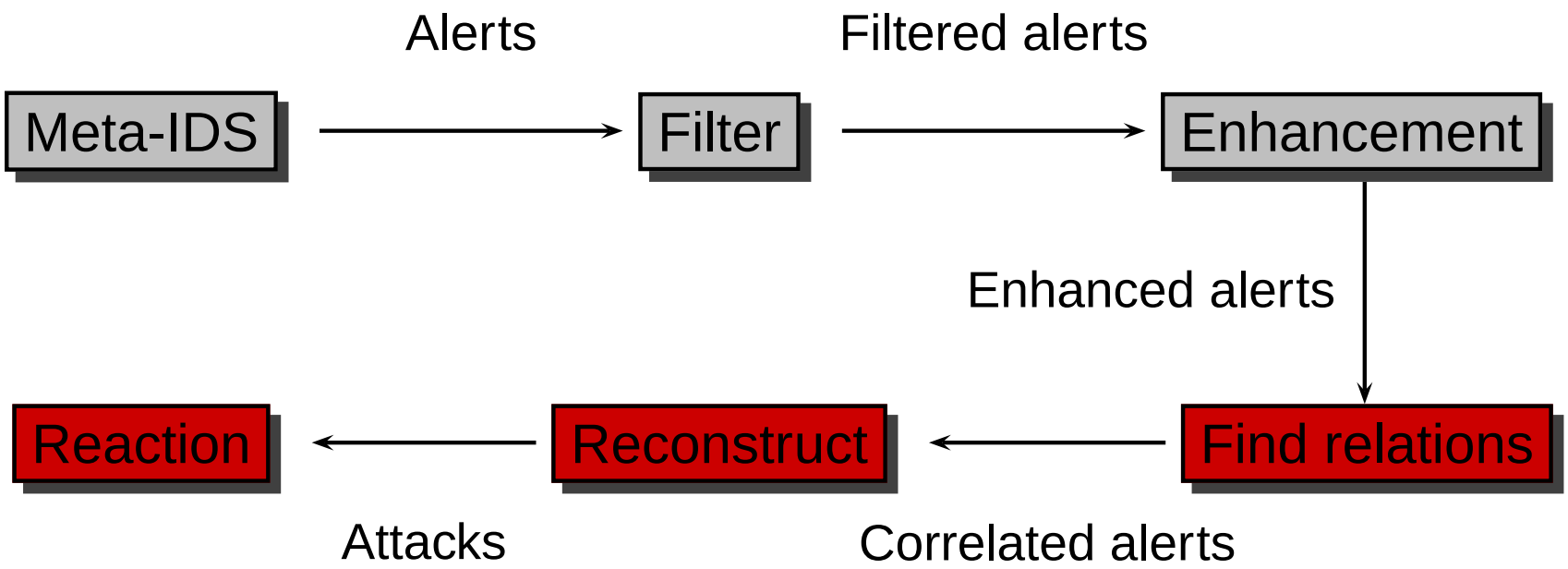


- Try to reconstruct the attack (events and timeline)
- Match vs patterns of known attacks

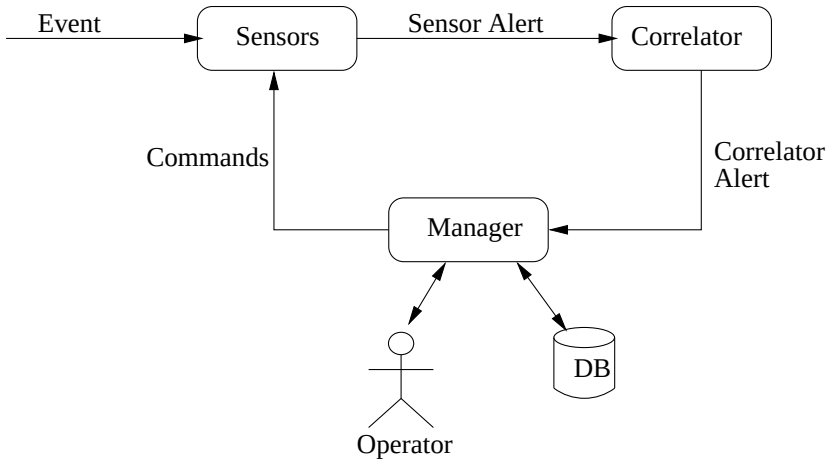
Find relations



Correlated Alert	Attack
Sequence: Scan + Incoming connection + Outgoing connection	Attack High success probability *known pattern*


Alerts
Filtered alerts
Meta-IDS
Filter
Enhancement
Enhanced alerts
Reaction
Reconstruct
Find relations
Attacks
Correlated alerts

Trust Score evaluation



- Attack is reconstructed and identified
- Trust Score is part of the decision to react
- Ability to capture the whole session by sending commands to agents

Reaction

- Report problem (mail)
- Archive
- Prepare a visualization
- Counter-measure
  - (try to) block attack (*dangerous !*)
  - Collect more information
  - Send commands to agents
- Notify

Pierre Chifflier, Sébastien Tricaud

INL - 101/103 Bvd MacDonald - 75019 Paris, France

IDS visualization

- Required to manage large amount of data
- Helps to focus on what is important
- Uses the human correlation engine
- Helps to write correlation signatures

Problem

- Alert are complex objects
- Numerous criteria (N-dimensional plot)
- How to graph correctly?

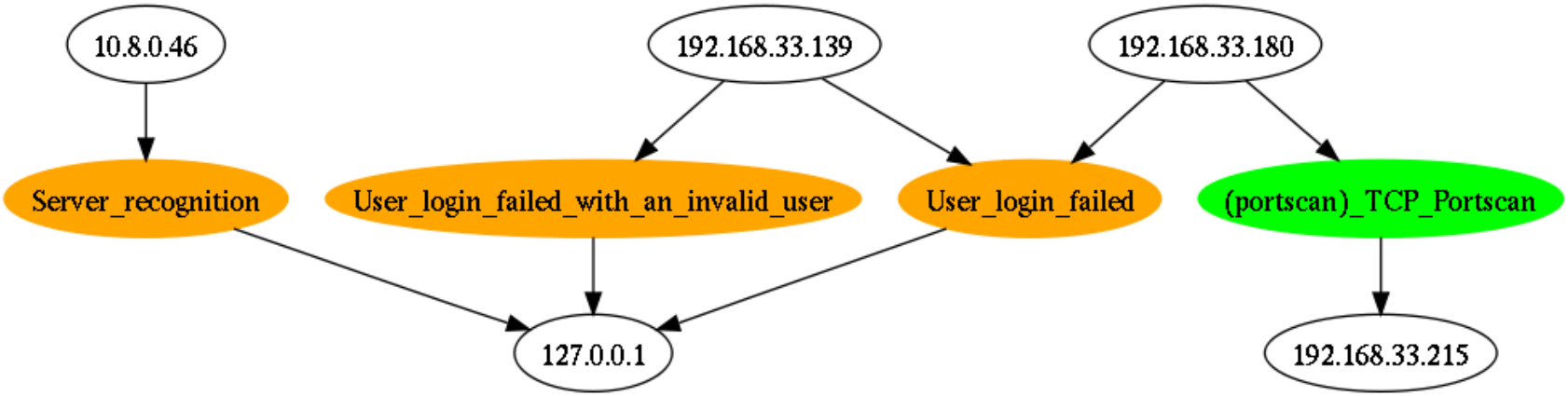
Visualization techniques

What we use:

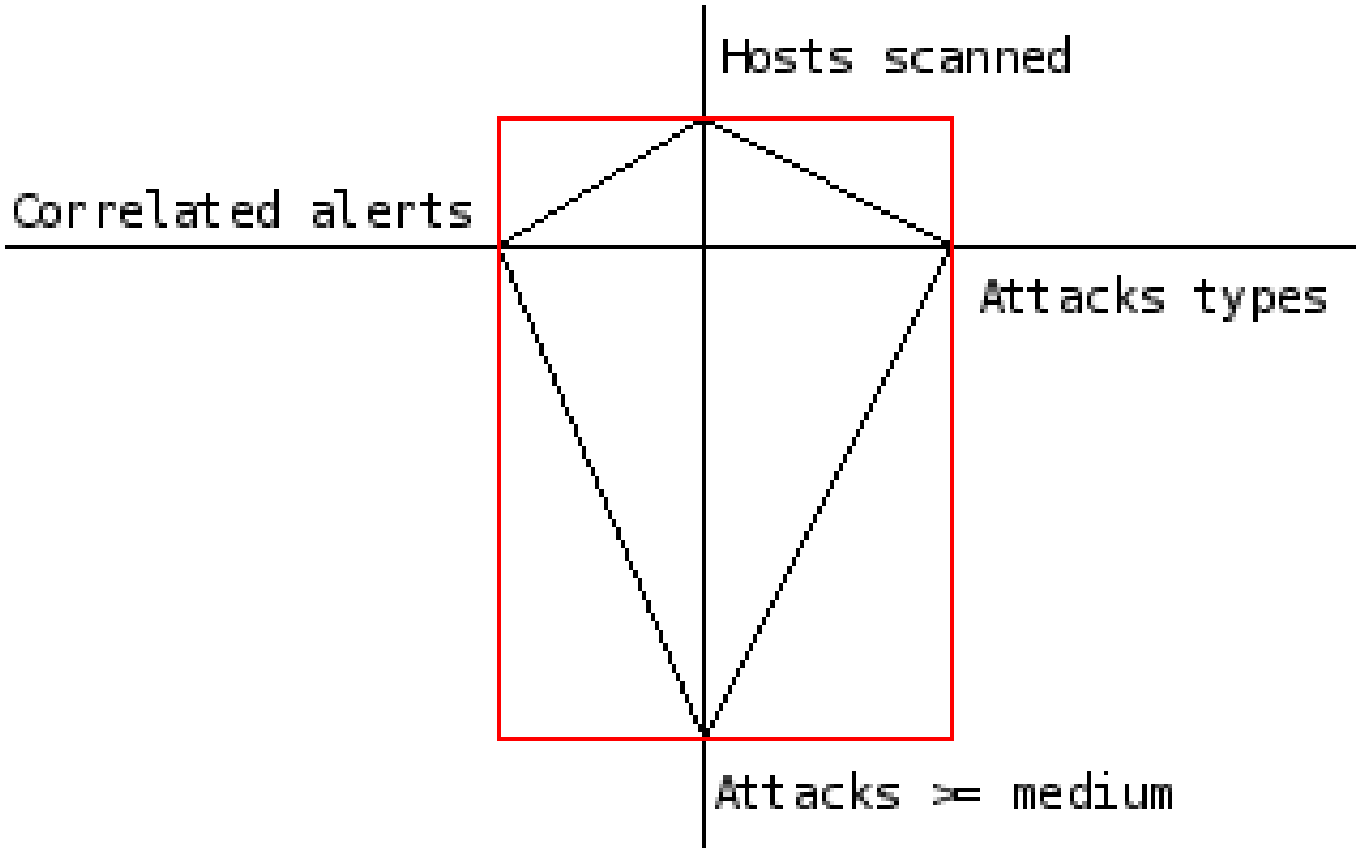
- Parallel coordinate plot
- 2D nodes
- 3D nodes
- Starplot
- Other (Treemap, . . .)

2D nodes

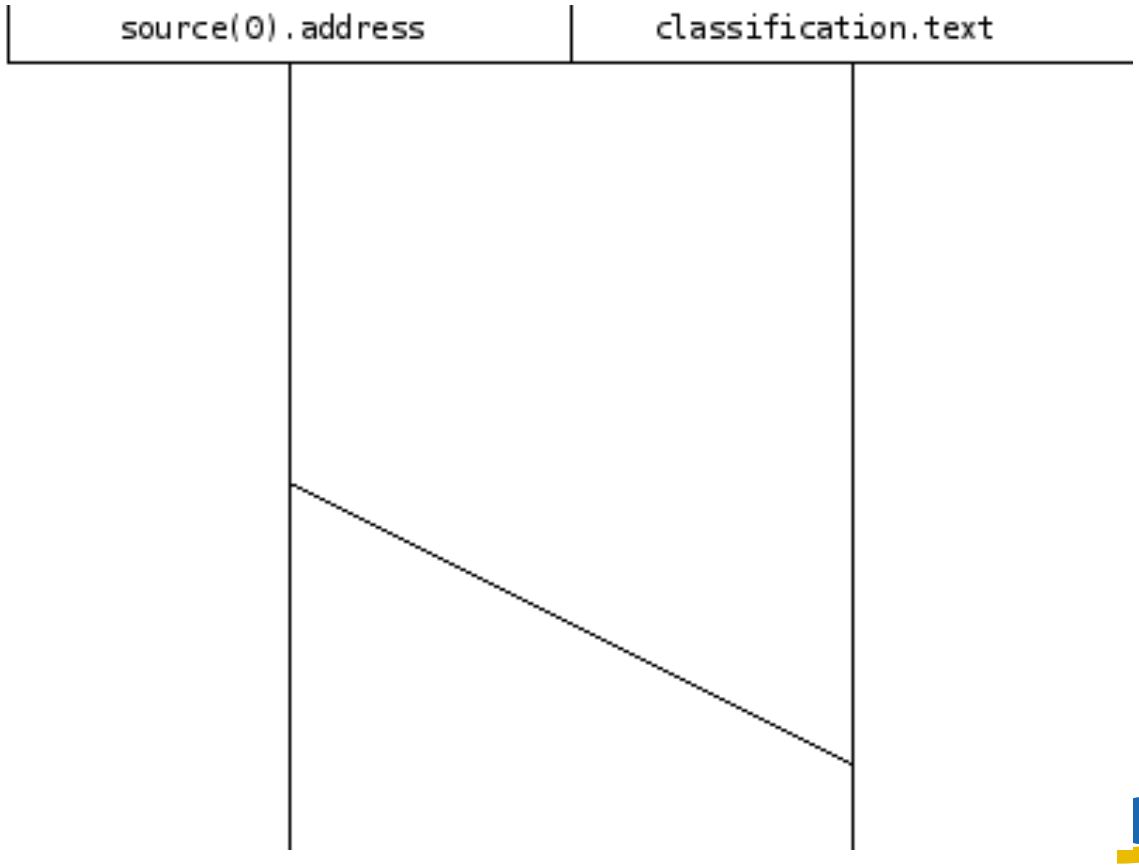
Graphviz makes easy to use relations



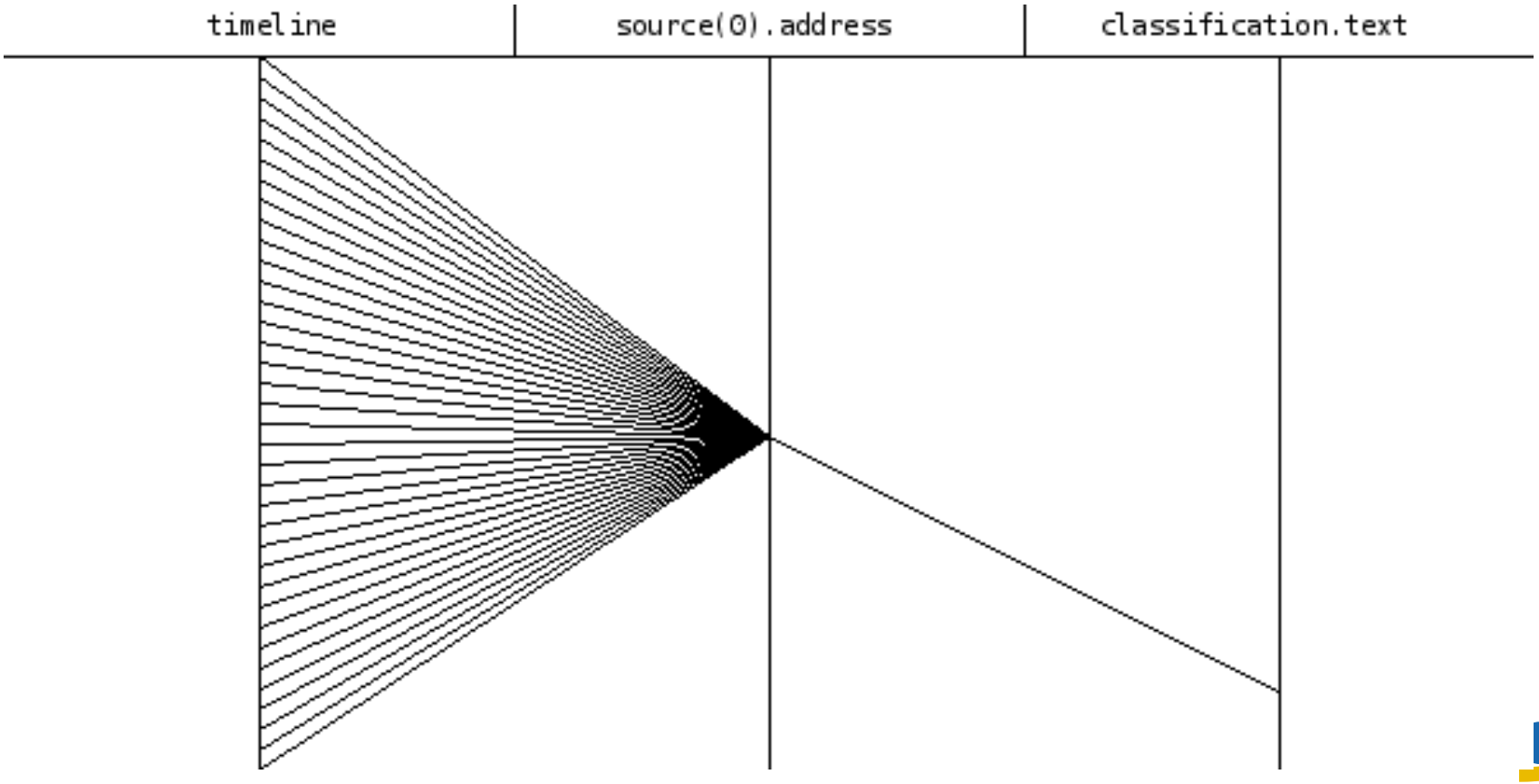
Starplot


Hosts scanned
Correlated alerts
Attacks types
Attacks >= medium

Visualization dilemma: take the right parameters for the right graph



Visualization dilemma: take the right parameters for the right graph



Relevant parameters from IDMEF paths

- Source (*alert.source(0).node.address(0).address*)
- Destination (*alert.target(0).node.address(0).address*)
- Impact (*alert.assessment.impact.severity*)
- Completion (*assessment.impact.completion*)
- Attack vector (*alert.classification.text*)
- Agent type (*analyzer(0).class*)

Code 1/3

- Based on Prelude IDS
- High-level language
- Python + Prelude Easy bindings

```
svn co http://svn.prelude-ids.org/libprelude/
   branches/libprelude-easy-bindings
```

Code 2/3

How to get alerts

```
from PreludeEasy import *

client = ClientEasy("pig", Client.IDMEF_READ)
client.AddConnection("192.168.33.215")
client.Start()
idmef = client.RecvIDMEF()
```

Code 3/3

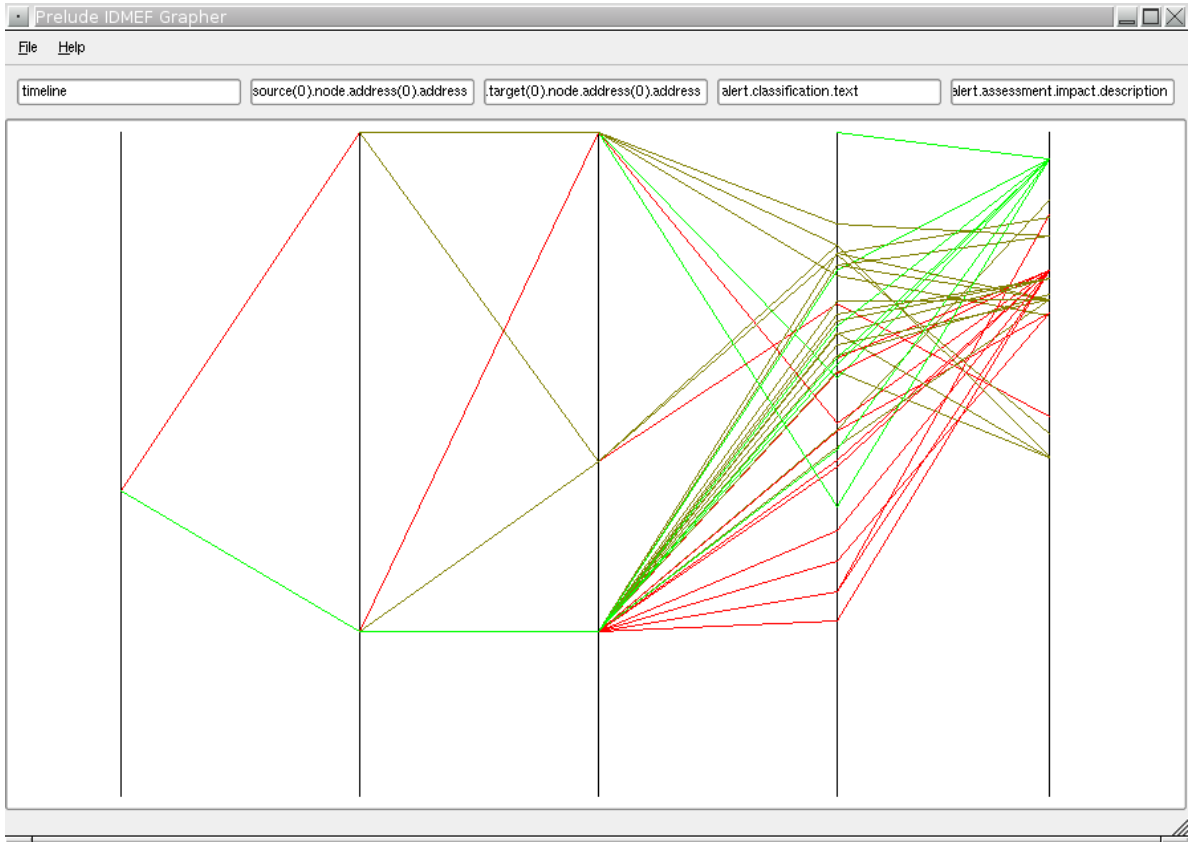
Graph Objects (GO!)

```
pen = QtGui.QPen()
pen.setColor(colorize_impact_severity(idmef))

line1_y = GetYPos(
      idmef.Get("alert.target(0).node.address(0).address"))
line2_y = GetYPos(
      idmef.Get("alert.classification.text"))

scene.addLine(
      line1_x, line1_y,
      line2_x, line2_y,
      pen)
```

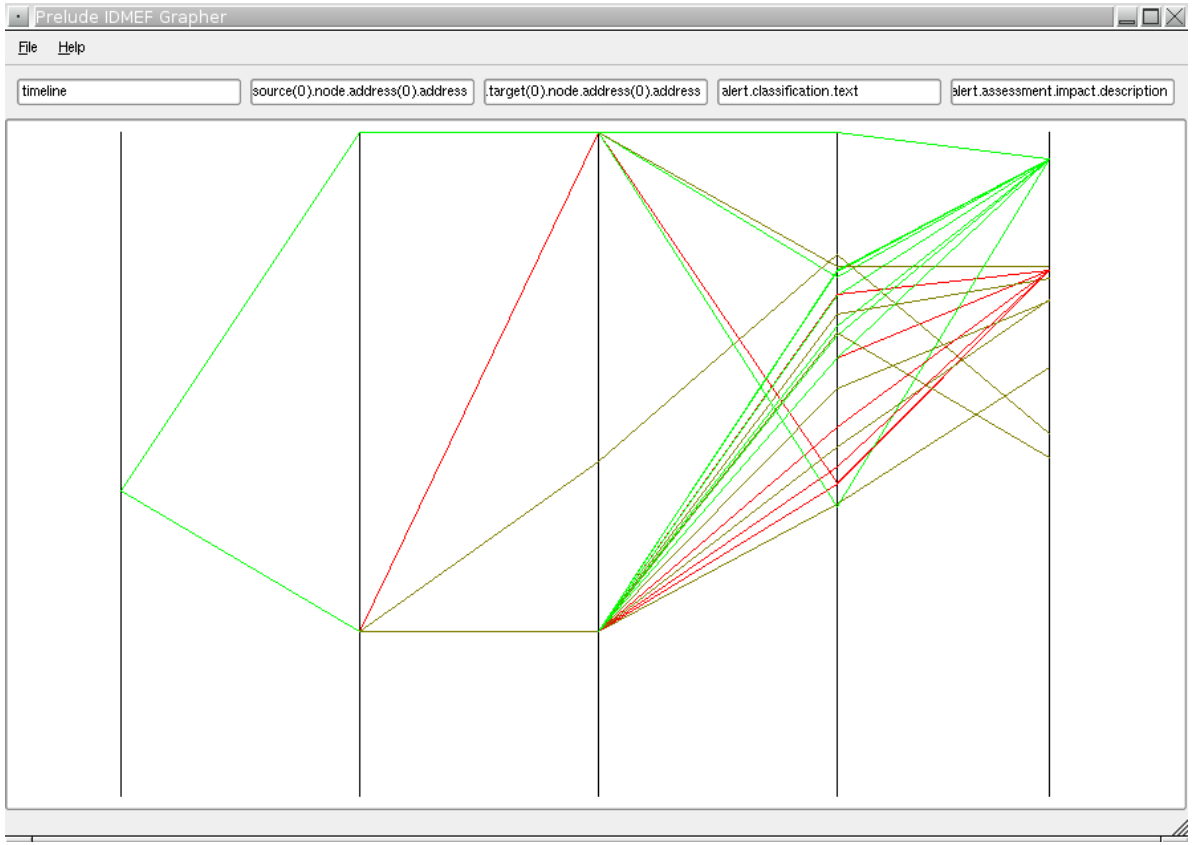
Prelude IDMEF Grapher (pig)

- Shows IDMEF paths
- Uses Prelude IDMEF pool
- Interesting to quickly understand a scanner
- Snort and LML are used as agents

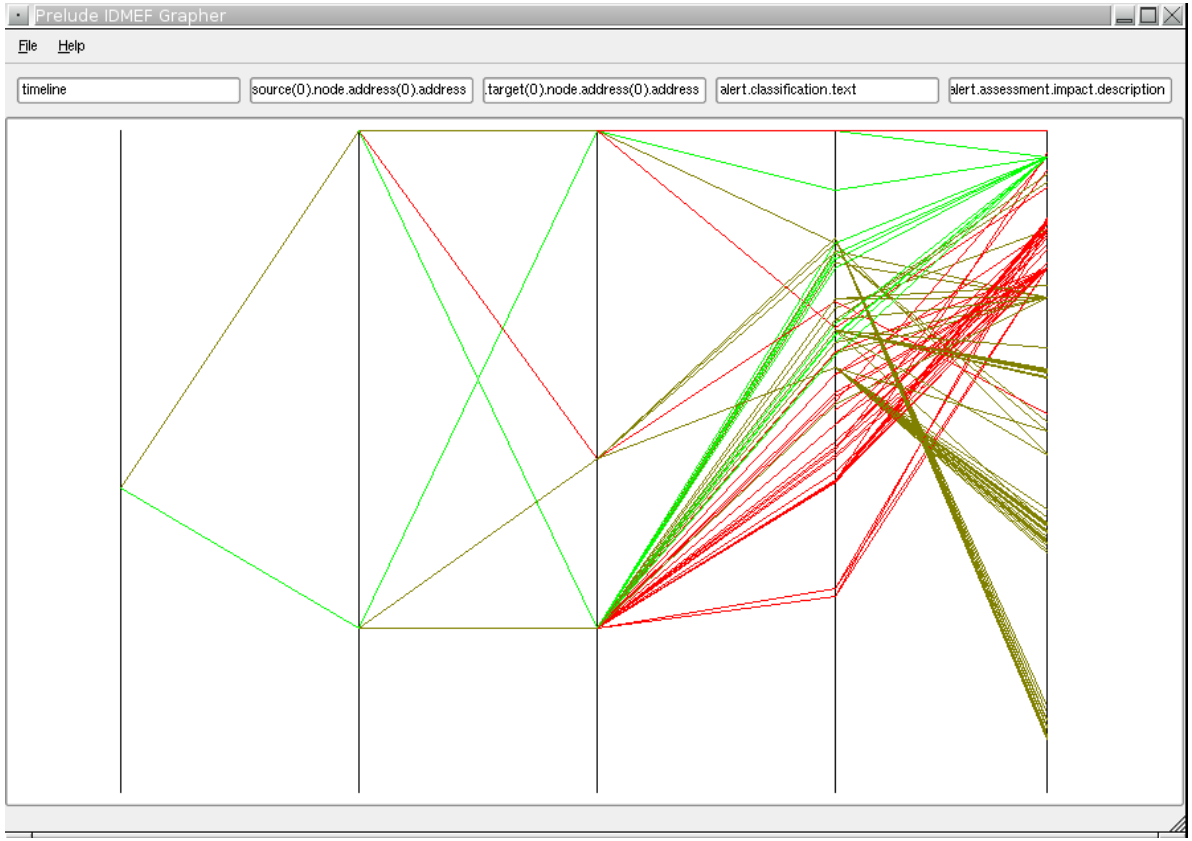
Saint: 166 alerts generated



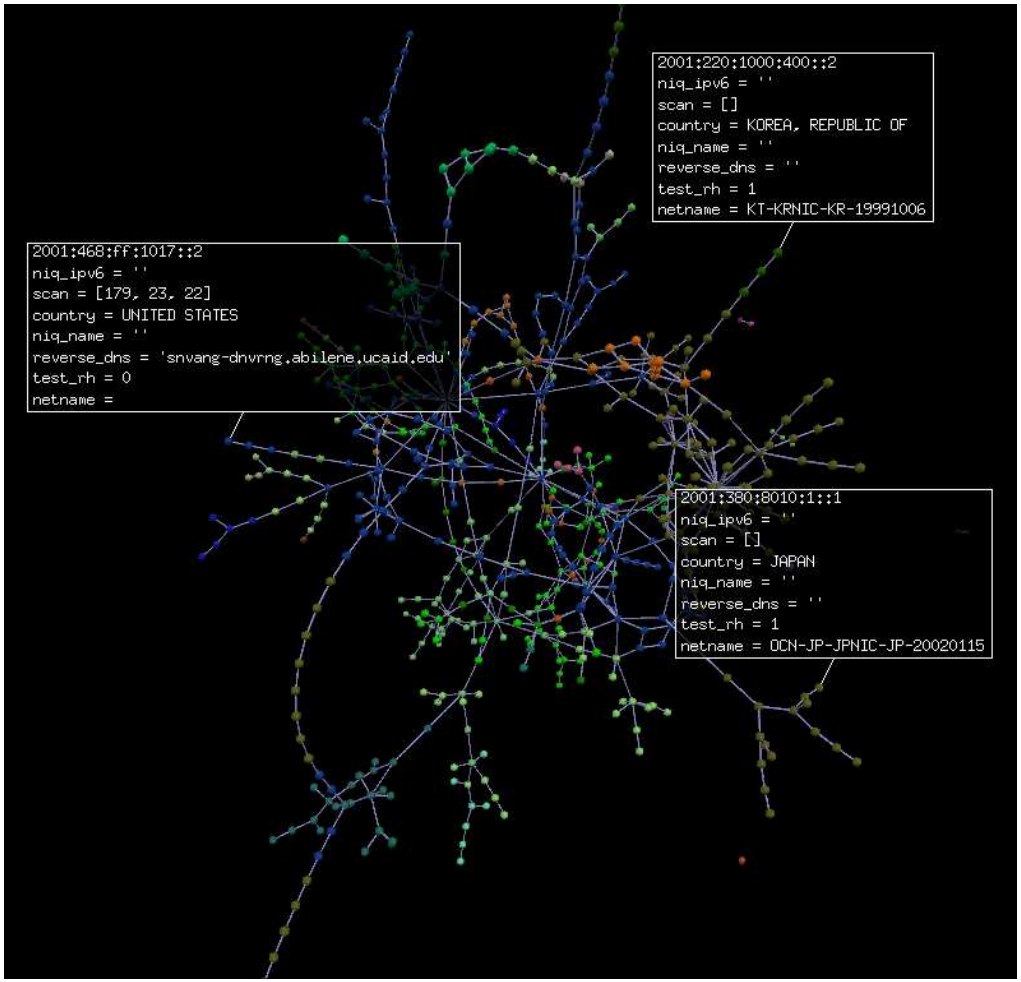
Retina: 76 alerts generated



Nessus: 1019 alerts generated

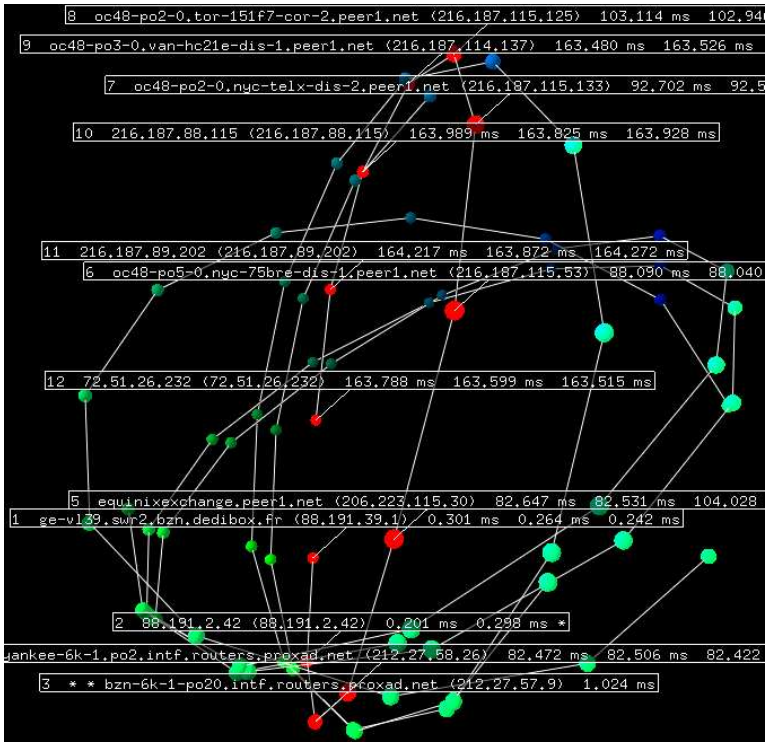


RTGraph3d


2001:220:1000:400::2
niq_ipv6 = ''
scan = []
country = KOREA, REPUBLIC OF
niq_name = ''
reverse_dns = ''
test_rh = 1
netname = KT-KRNIC-KR-19991006
2001:468:ff:1017::2
niq_ipv6 = ''
scan = [179, 23, 22]
country = UNITED STATES
niq_name = ''
reverse_dns = 'snvang-dnvrng.abilene.ucaid.edu'
test_rh = 0
netname =
2001:380:8010:1::1
niq_ipv6 = ''
scan = []
country = JAPAN
niq_name = ''
reverse_dns = ''
test_rh = 1
netname = OCN-JP-JPNIC-JP-20020115

GraphGL

We were jealous of rtgraph3d ;-)



Available at *http://www.dindinx.net/graphgl/*

Wolfotrack: Netfilter connection tracker made easy

Visualization Pros and Cons

	Starplot	2D	3D	Parallel Coordinate Plot
Large number of alerts	No	No	Yes	Yes
Large number of criteria	No	Yes	Yes	Yes
Time base representation	No	No	No	Yes
Easy to read	No	Yes	Yes	Yes
Live filtering	No	No	Yes	Yes

Summary

- Visualization is still under construction
- Until now, parallel multi-axes view is the best we've found
- We still do not know the best view for the best criterion
- There is not just one good visualization

Future work

- Understand application layer better
- For how long should we monitor an attack ?
- Write more correlation rulesets
- Find better visualization models

Acknowledgment

- INL staff
- Yoann Vandoorselaere
- Philippe Saadé
- David Odin
- RV Martin
- Elodie and Anthony

Questions ?

Thank you for your attention

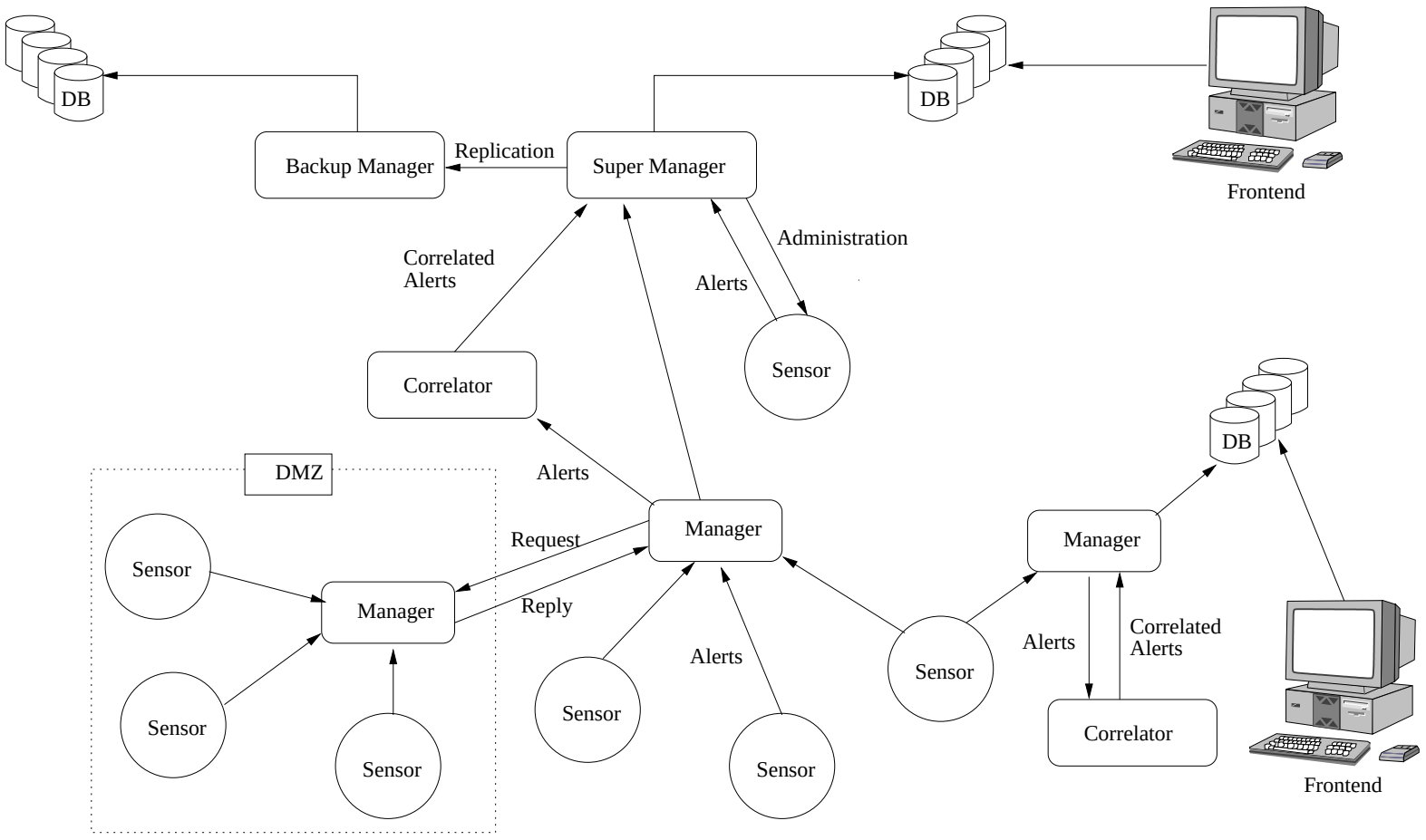
Contact us !

- Pierre Chifflier <p.chifflier@inl.fr> https://www.wzdftpd.net/blog
- Sébastien Tricaud <s.tricaud@inl.fr> http://www.gscore.org/blog
- INL http://www.inl.fr
- Prelude IDS http://www.prelude-ids.org
- Prelude IDS Trac http://trac.prelude-ids.org

References

- Conti, G.: Security Data Visualization. No Starch Press, San Francisco, CA, USA (2007)
- Valdes, A., Skinner, K.: Probabilistic alert correlation. Lecture Notes in Computer Science 2212 (2001)
- Schultz, E.: Intrusion detection event correlation: Approaches, benefits and pitfalls (March 2007) CERIAS Security Seminars.

Prelude user architecture



Example: NuFW

Example of agent: NuFW (`http://www.nufw.org`)

- **authenticating** firewall, based on user identity
- Provides a native Prelude module for log
- Add information on users on each connection
- Add valuable information for correlation
- Allows to strictly apply the Security Policy

Example of alert: NuFW (1)

- Example of IDMEF alert, with interesting fields.
- Alert emitted for a new HTTP connection using Firefox.

```
messageid: 5478076470
analyzer(1):
        analyzerid: 2334565015741136
        name: nufw
        manufacturer: http://www.nufw.org/
        model: NuFW
        version: 2.3.0 ($Revision: 3475 $)
        class: Firewall
        ostype: Linux
        osversion: 2.6.20-15-386
        process:
                name:
                pid: 15197
```

Example of alert: NuFW (2)

```
create_time: 29/06/2007 11:26:24.0 +02:00
classification:
        text: Connection opened
detect_time: 29/06/2007 11:32:56.0 +02:00
analyzer_time: 29/06/2007 11:32:56.642005 +02:00
source(0):
        spoofed: unknown (0)
        node:
                category: unknown (0)
                address(0):
                        category: ipv4-addr (7)
                        address: 192.168.0.2
        user:
                category: application (1)
                user_id(0):
                        type: current-user (1)
                        name: pollux
                        number: 1000
        process:
                name: firefox
                path: /usr/bin/firefox
        service:
                iana_protocol_number: 6
                iana_protocol_name: tcp
                port: 3489
```

Example of alert: NuFW (3)

```
target(0):
        decoy: unknown (0)
        node:
                category: unknown (0)
                address(0):
                        category: ipv4-addr (7)
                        address: 82.165.85.221
        service:
                iana_protocol_number: 6
                iana_protocol_name: tcp
                port: 80
assessment:
        impact:
                severity: low (2)
                type: user (5)
                description: Connection state changed
```

Our attack classification:

- Authentication
  - Local user
  - System user
  - Admin user
  - Other
- Probe
  - Protocol
  - Scan
  - Sniff
  - Users
  - Other
- Corruption
  - File
  - Application
  - Other
- Availability (Denial of Service)
  - Resource consumption
  - User account locking
  - Application crash
  - Other

Our classification

- The alert itself is *not* sufficient to find the category
- **Use the alert (low-level), correlation, to find the type (category) of the attack**
- No global catch-all category (one per section)
- **clear separation between the goal and the type**
- Don't mix the goal with the type of the attack: A file corruption may be used for Probe as well as for Penetrate (the same exploit is often used for Probe and Penetrate)
- We group attack means in each defined Goal