Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Collaborative Incident Handling Based on the Blackboard-Pattern

Nadine Herold, Holger Kinkelin and Georg Carle

November 8, 2016

Chair of Network Architectures and Services
Department of Informatics
Technical University of Munich

Contents

Motivation and Background

Related Work

Problem Statement

System Design and Implementation

Evaluation

Conclusion

Motivation and Background

Related Work

Problem Statement

System Design and Implementation

Evaluation

Conclusion

Motivation

- Amount and variants of attacks on networks is growing
- Defending networks manually is impossible
- Automated incident handling is highly beneficial
  - Continuously defend the network
  - Respond quickly
  - Less error-prone
  - Systematical incident response
- We focus on intrusion handling

Background: Typical Intrusion Handling Steps

- Network Monitoring (NMS) and Intrusion Detection Systems (IDS) collect information about the network and its healthiness
  - NMS: collect infrastructure information
  - IDS: raise alerts when an intrusion is detected
- Alert Processing Systems (APS) aggregate, correlate and prioritize alerts
  - Gain more insights into the intrusion by analyzing the situation
- Intrusion Response Systems (IRS) counteract automatically
  - Identify suitable responses
  - Execute reponses on the target network, e.g., block a rogue host

Motivation and Background

Related Work

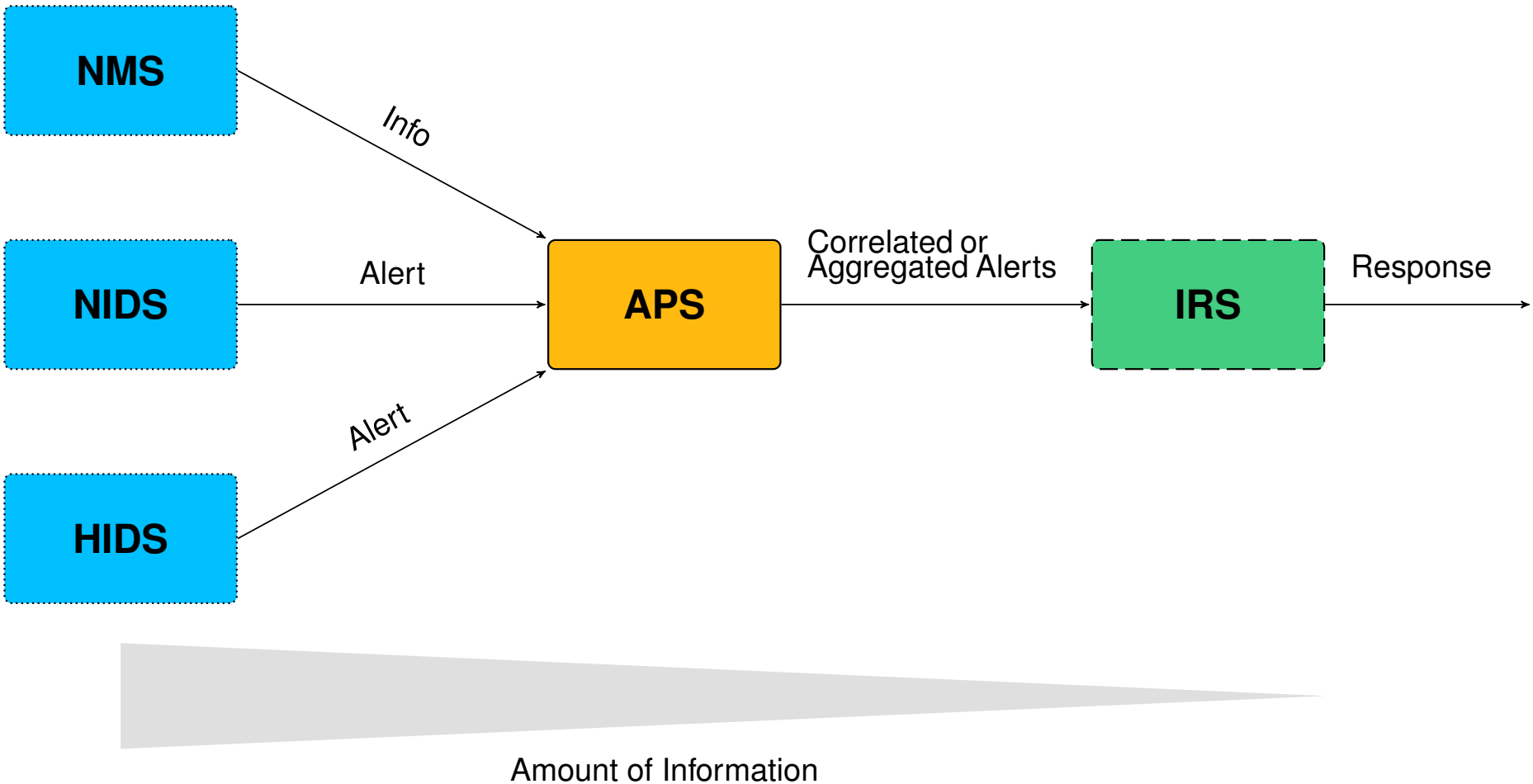
Problem Statement

System Design and Implementation

Evaluation

Conclusion

Execution Model: Pipelined Intrusion Handling



Other Execution Models

- Pipelined intrusion handling
  - Information loss from step to step
  - Limited information sharing capabilities
- Intrusion handling using Complex Event Processing (CEP)
  - Window size difficult to determine
    (too large $\rightarrow$ low performance; too small $\rightarrow$ information loss)
  - Limited information sharing capabilities
- Agent-based systems for intrusion handling
  - Central intelligent master component needed to dispatch information to agents

Motivation and Background

Related Work

Problem Statement

System Design and Implementation

Evaluation

Conclusion

Problem Statement

- Significant effort has been made to improve each intrusion step individually
- No solution exists that interleaves steps and creates a comprehensive view on the target network
  - Information already collected/computed in previous steps is lost for being used by subsequent steps
  - Information and intermediate results cannot be shared efficiently between single steps
- Post-incident forensics of intrusion handling activities difficult

Motivation and Background

Related Work

Problem Statement

System Design and Implementation

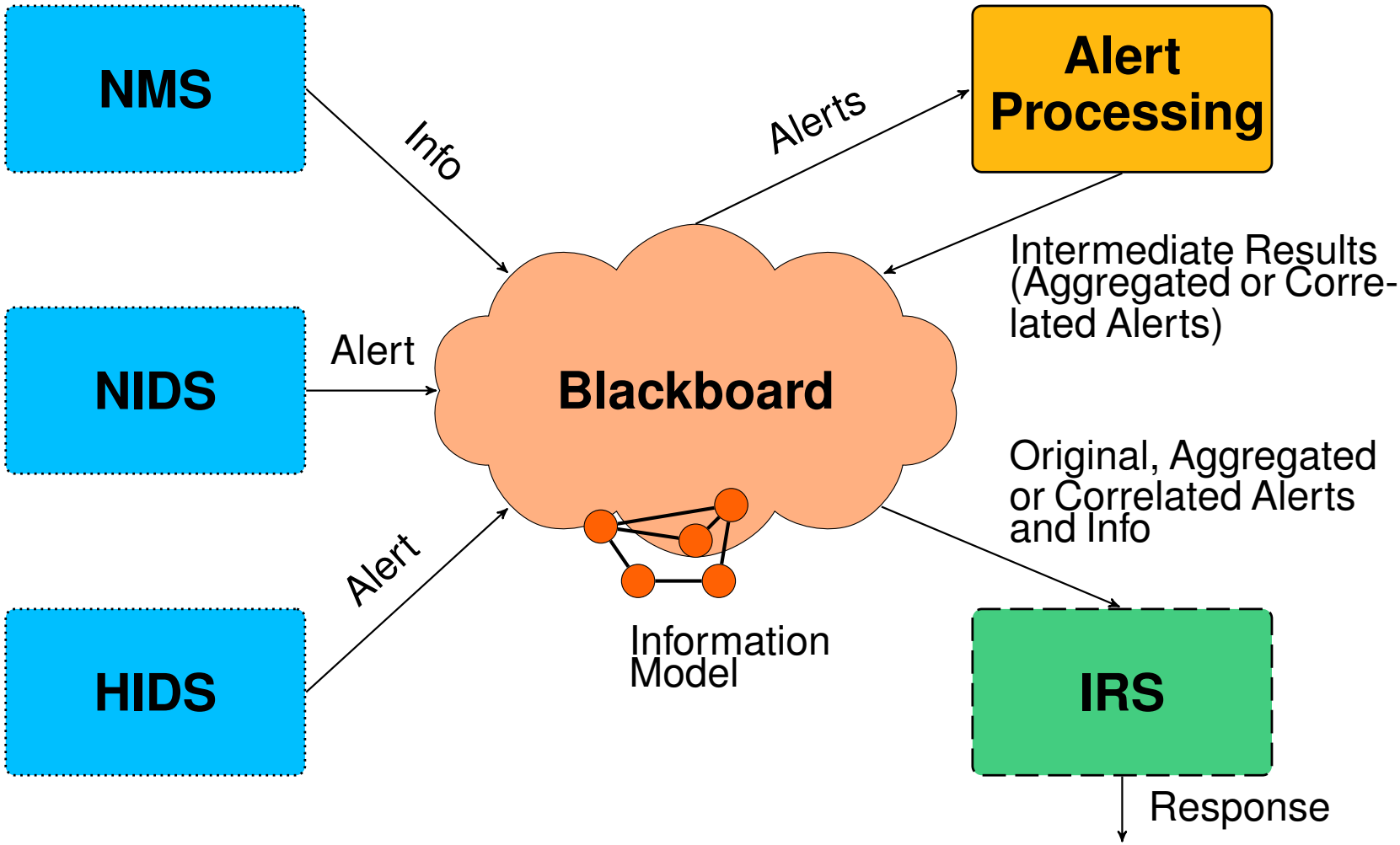
Evaluation

Conclusion

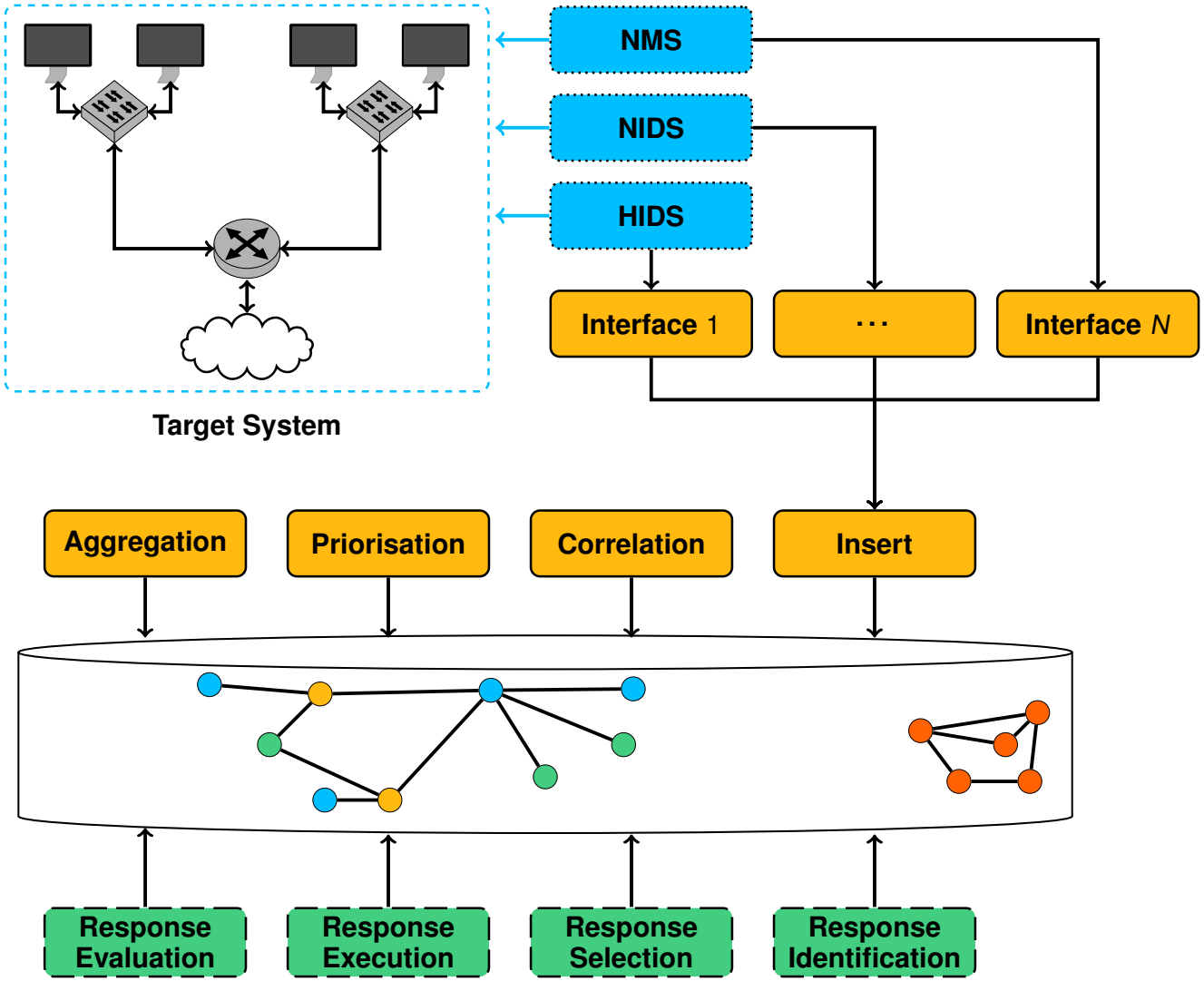
Introducing the Blackboard Pattern

- The blackboard pattern is applicable to problems that can be decomposed into smaller sub-problems / sub-tasks
  - Example: (distributed) incident handling / intrusion handling
- Sub-tasks solve their sub-problem and share their intermediate results with other sub-tasks
- Original information remains untouched
- Original information + intermediate results can be reused by sub-tasks to further tackle the problem
- Blackboard needs an Information Model specifically designed for the problem domain

Blackboard-based Intrusion Handling



System Overview



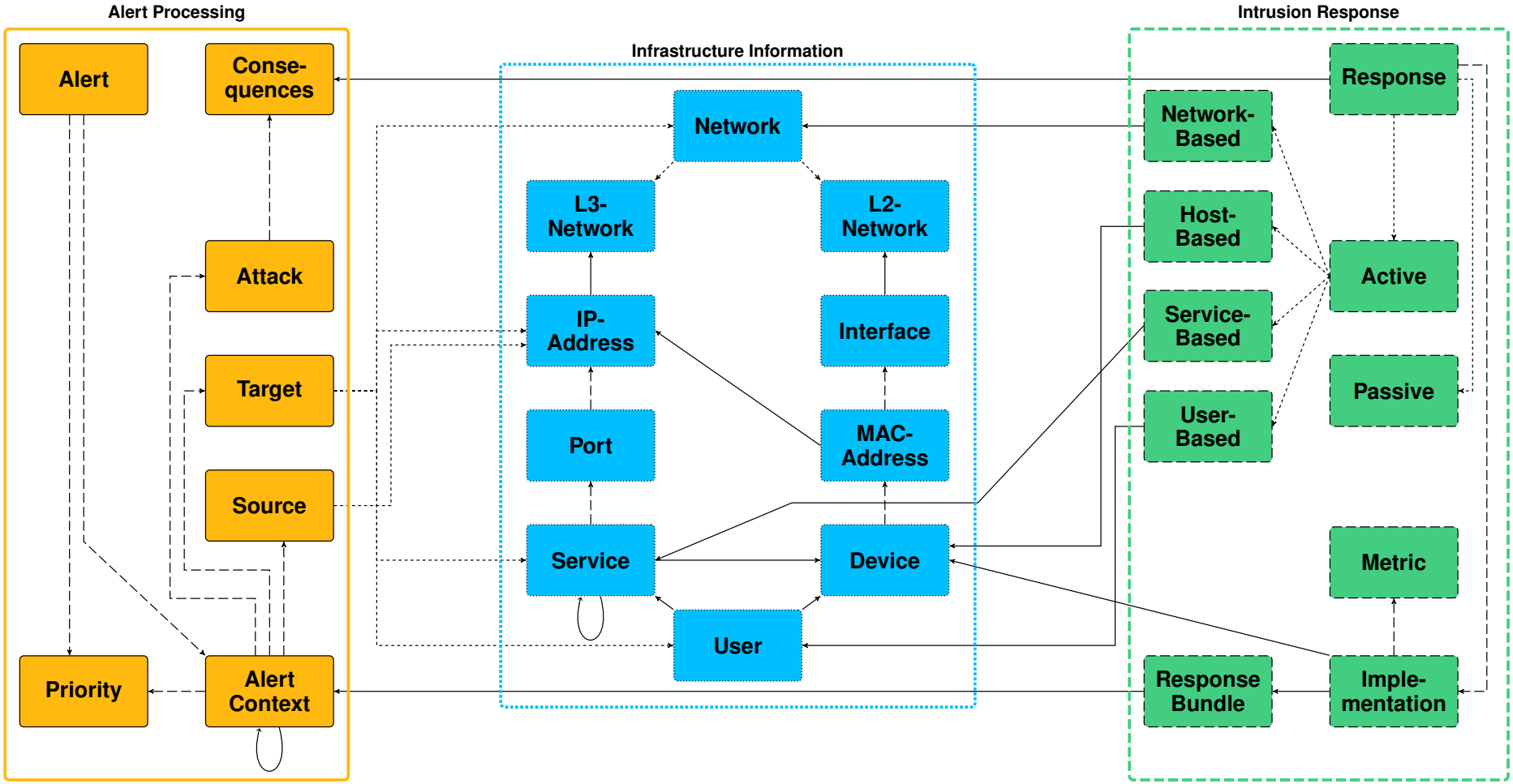
Requirements on an Information Model

... suitable for intrusion handling

- R1: Separation – Segmentation of information enables updating/adding of information by different modules
- R2: Completeness – Information for all steps of Incident Handling needs to be present
- R3: Compatibility to the IDMEF standard[1] used by many IDSes

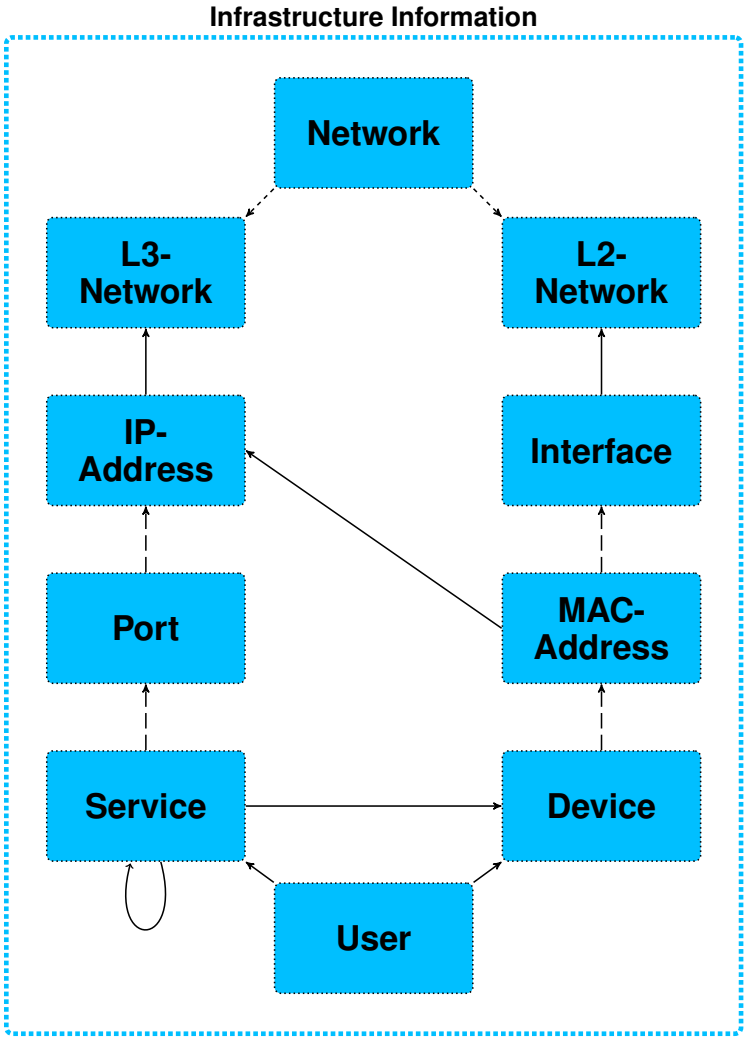
[1] Intrusion Detection Message Exchange Format, RFC 4765

Information Model for Intrusion Response - Overview



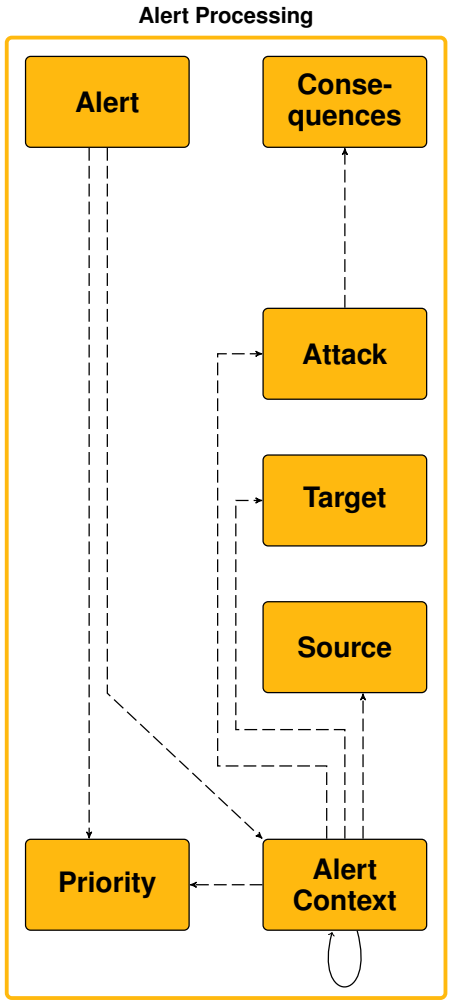
Infrastructure Information Model – Examples

- NMSes send their scanning results to specific interfaces which add the info to the Blackboard
- A `Service` runs at a `Port` opened on a NIC with an `IP-Address` belonging to a `L3-Network`
- A `Device` has a `NIC` with `MAC-Address` and assigned `IP-Address`
- A `User` is logged into `Device`
- A `User` uses `Service`



Alert Information Model – Examples

- IDSes send IDMEF messages containing alerts to specific Blackboard Interfaces
- IDMEF alerts are normalized and combined into an `Alert Context`
  - `Source` (of attack)
  - `Target` (of attack)
  - `Attack` (type)
- `Alert` and `Alert Context` nodes have a `Priority`



Implementation

- Python 3
- Object oriented implementation of Information Model
- Automatic translation of class structures to suitable database design
- Two different databases/database types used:
  - Relational: postgreSQL
  - Graph-based: OrientDB

Motivation and Background

Related Work

Problem Statement

System Design and Implementation

Evaluation

Conclusion

Evaluation – Test Data Sets and Test Cases

→ Measure the prototype's performance under varying conditions

- Test data sets simulate different attacks:

DDoS DDoS: many sources attack a small number of targets
AP Attack path: an attack spreads in the network
F Flooding: Mulitple IDSes raise the same alert

- Test data set size: from 1000 to 5000 alerts

- Test cases simulate typical tasks of the intrusion handling system

ins Node Insertion – Adding of `Alert` and `Alert Context` nodes
prio Node Prioritization – Updates `Priority` attribute of `Alert` and `Alert Context` nodes with random number
comb Node Combination – Combining related `Alerts Context` nodes

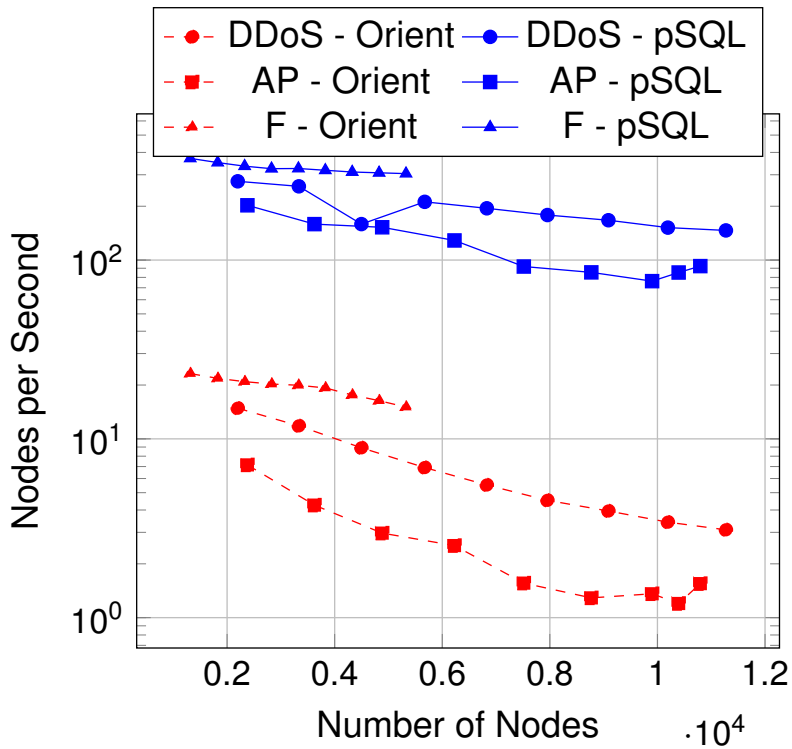
- Test cases are cumulative, e.g., t3 contains t1 and t2

Measurement Results: Alerts per Second

Exp.	$pSQL_{min}$	$pSQL_{max}$	$pSQL_{avg}$	$Orient_{min}$	$Orient_{max}$	$Orient_{avg}$
$DDoS_{ins}$	287.09	354.72	320.75	11.4	19.72	14.73
$DDoS_{prio}$	228.61	307.27	257.8	8.4	16.24	11.55
$DDoS_{comb}$	64.97	125.44	86.15	1.37	6.75	3.12
AP_{ins}	299.4	355.76	324.76	12.5	19.35	15.13
AP_{prio}	230.36	287.86	250.71	8.91	16.23	11.62
AP_{comb}	30.80	85.12	49.59	0.51	3.01	1.1
F_{ins}	370.32	396.63	384.58	37.88	50.87	44.77
F_{prio}	318.1	330.31	325.04	15.4	35.29	23.38
F_{comb}	281.78	293.31	287.73	14.13	18.00	16.97

Table contains min, max and average rates of all test data set sizes

Measurement Results: Nodes per Second



Graph shows results of node combination test case

Motivation and Background

Related Work

Problem Statement

System Design and Implementation

Evaluation

Conclusion

Conclusion

- Related work has drawbacks: information sharing is difficult between intrusion handling steps, information loss, ...
- **Our contributions:**
  - Blackboard-pattern for intrusion handling
  - Suitable information model
  - $\rightarrow$ Enables Information sharing between intrusion handling steps
  - Proof-of-concept implementation using two different DBs
- Future Work:
  - Information security of the data on the Blackboard
  - Improving performance

Contact

Thank you for the audience!

Nadine Herold, Holger Kinkelin and Georg Carle

Technische Universität München
Department of Informatics
Chair of Network Architectures and Services
Boltzmann Straße 3
85748 Garching bei München
Germany

{lastname}@net.in.tum.de
https://github.com/Egomania/BlackboardIDRS