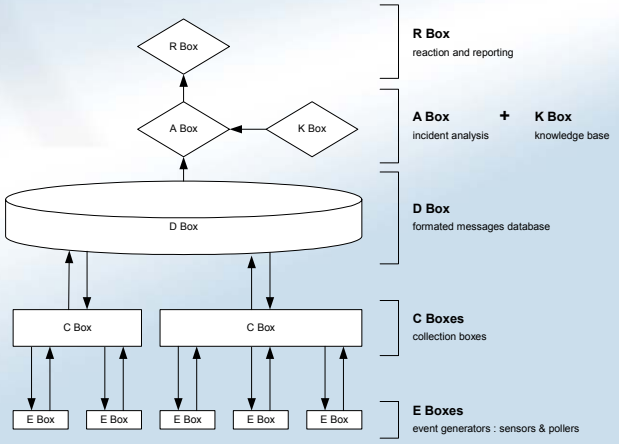
Security Operation Center

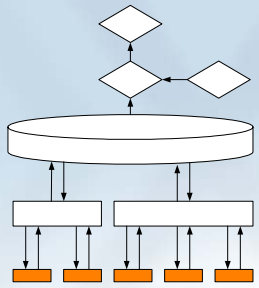
Concepts and Implementation

renaud.bidou@intexxia.com

- SOC Modules
- Global Architecture
- Collection & Storage
- Correlation

> SOC Modules





> SOC Modules

> E Boxes

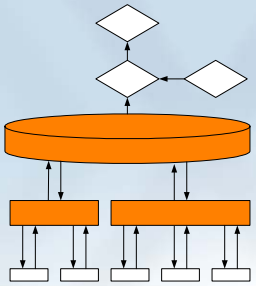
- event generation
- passive : sensors
- active : pollers

> Sensors

- IDS, filtering eq., syslog, apps, honeypots …
- running in hostile environment
- lack of standard for host-based sensors

> Pollers

- third-party tool
- status evaluation
- may encounter performance problems



> SOC Modules

> C & D Boxes

- event collection & storage
- standard formating

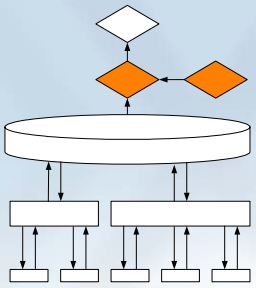
> Collection

- set of multi-protocol / application agents
- lack of standard format
- availability and performance concerns

> Storage

- duplicates merging
- performance concerns with huge volume of events

intexio



> SOC Modules

> A & K Boxes

- multi-level analysis
- intrusion scenarii
- system status

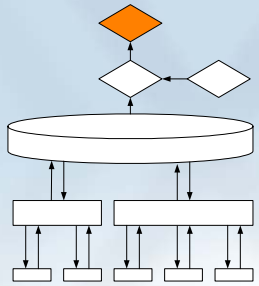
> Analysis & Correlation

- heavy research focus
- proof of concept implementation
- proprietary technologies

> Knowledge Base

- vulnerabilities & intrusion scenarii
- system security status
- security policy

intexio



> SOC Modules

> R Boxes

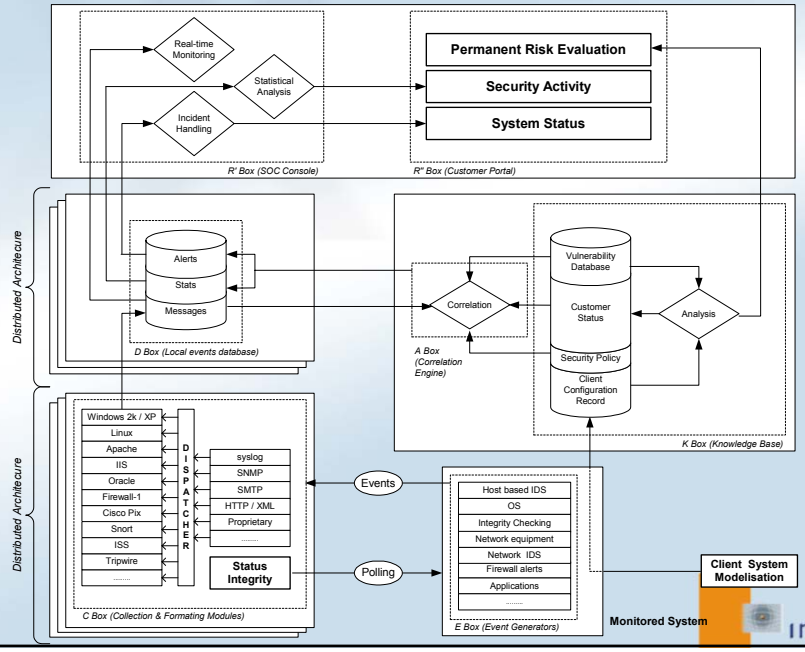
- reaction & reporting
- operators interfaces
- end-user interfaces

> Interfaces

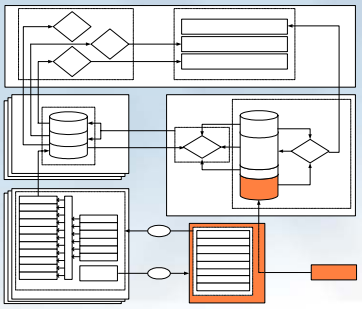
- subjectivity
- relies on best-practices and experience return
- MANDATORY

intexio

> Global Architecture



intexio



> Global Architecture

> Data acquisition

- technical inventory
- security policy review

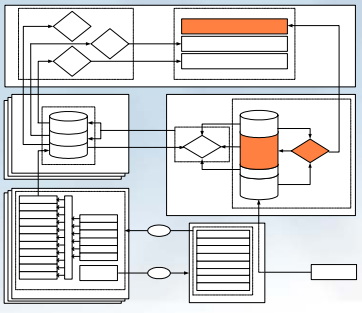
> Technical reviews

- intrusive & non-intrusive data acquisition techniques
- need for attack taxonomy and classification
- relative vulnerability impact

> Organizational reviews

- acceptable behavior definition
- access rights
- permitted operations

intexio



> Global Architecture

> Status Evaluation

- vulnerabilities definition
- security level evaluation
- permanent audit

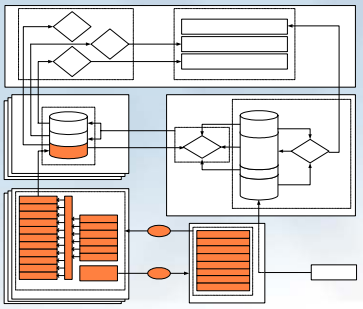
> Vulnerability database

- structural vulnerabilities
- functional vulnerabilities
- topology-based vulnerabilities

> Permanent security evaluation

- attack trees generation
- new evaluation performed when KB updated
- history management

intexio



> Global Architecture

> Events management

- generation
- collection
- formating & storage

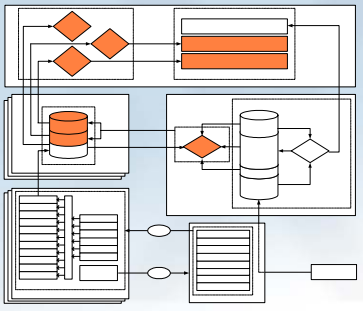
> Exhaustivity vs. performance

- events overload
- structural & policy pre-filter
- difficulty to manage distributed filters

> Collection and storage

- protocol agents
- source type identification
- message formatting

intexio



> Global Architecture

> Analysis & reporting

- event correlation
- operational reporting
- strategic reporting

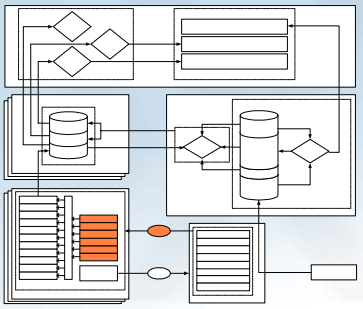
> Alerts

- structural and behavior alert generation
- criticity handling
- statistical analysis

> Interfaces

- operators consoles
- debugging consoles
- end-user portal

intexio



> Collection & Storage

> Data collection

- heterogeneous sources
- scalable architecture

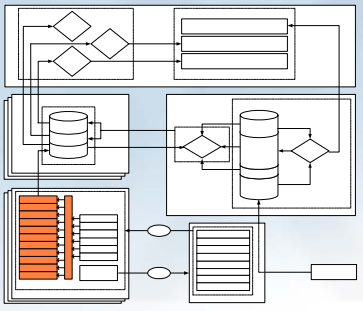
> Protocol agents

- server-side agents dedicated to one protocol
- multiple forwarding channels support
- no shared data = easy clustering / farming

> Reliability & security

- TCP encapsulation
- collection channel encryption

intexio



> Collection & Storage

> Data collection

- source sensor identification
- « standard » formatting

> Dispatcher

- pattern-based analysis
- forwarding to dedicated application agent
- multiple listening and forwarding channels support

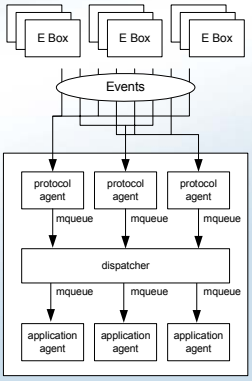
> Application agents

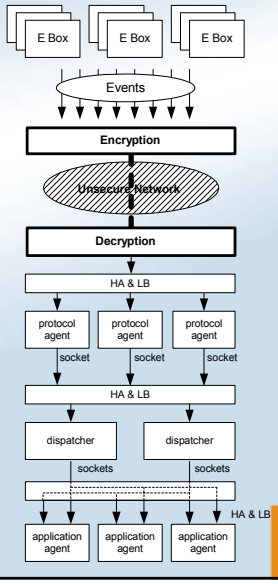
- dedicated to specific (sensor, Xmit protocol)
- message formating
- may be merged with dispatcher

intexio

> Collection & Storage

> Sample architectures

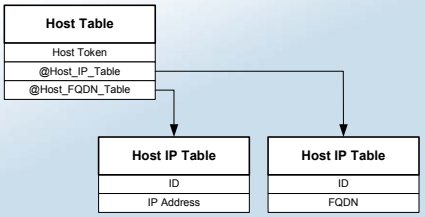


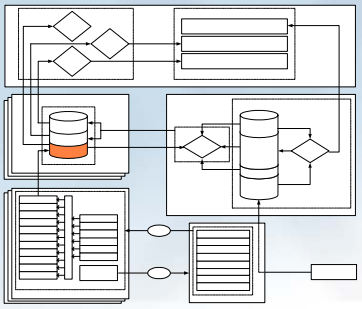


> Collection & Storage

> Host Entry

- unique host identification



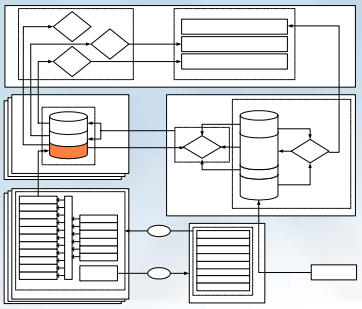


> Identification

- by IP
- by FQDN
- unique host token

> Needed to support

- multihoming
- NAT & Virtual IP
- virtual servers

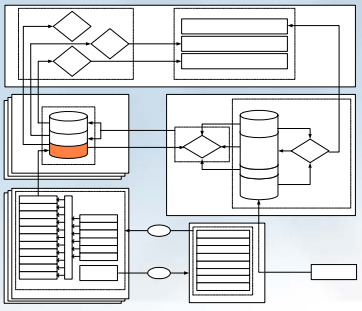


> Collection & Storage

> Messages format

- basic message formatting
- correlation ready

Field	Attributes	Description
id	Unique	Unique message ID
sensor_id	Not Null	Unique Sensor ID
msg_type	Not Null	Type of message (ipchains, snort-1.8.x-alert etc.)
epoch_time	Not Null	Date in epoch format of event generation
source		Intrusion Source Host Token
target		Intrusion Target Host Token
proto		Protocol number
src_port		Intrusion source port number
tgt_port		Intrusion target port number
info		Additional info
int_type_id	Not Null	Intrusion type ID (Filter, Access etc.)
int_id		Intrusion ID
message	Not Null	Original message



> Collection & Storage

> 3rd Party info

- additional information

> Sensor & Sensor Type tables

- sensor identification

> Message Type table

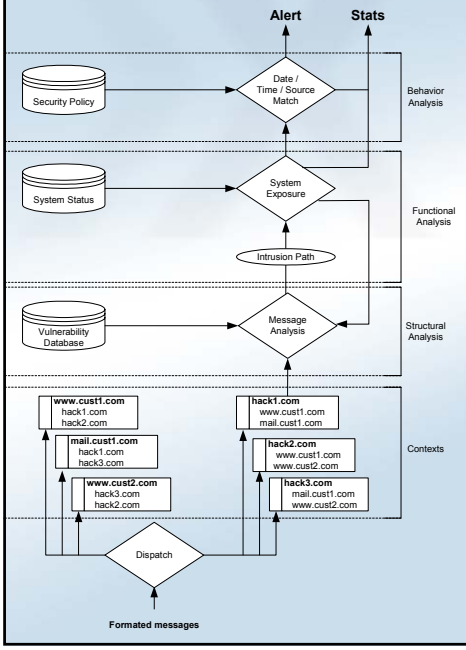
- human readable message type description

> Intrusion & Intrusion Type tables

- intrusion identification
- matches between different references

> Correlation

> Overview

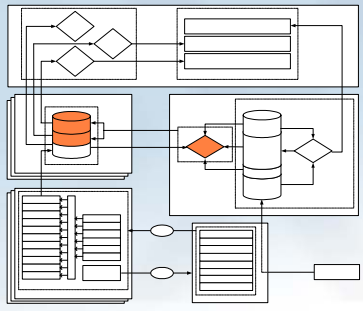


- duplicate identification
- sequence pattern matching
- time pattern matching
- system exposure & criticity
- security policy matching

intexio

> Correlation

> Contexts



- event grouping
- correlation preparation

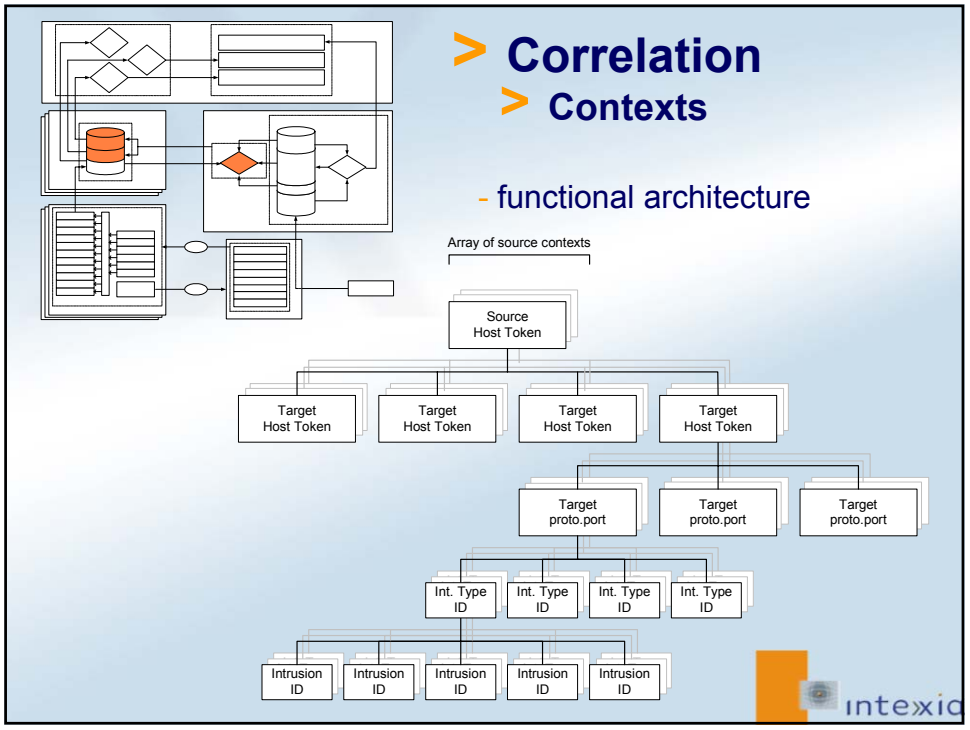
> Definition

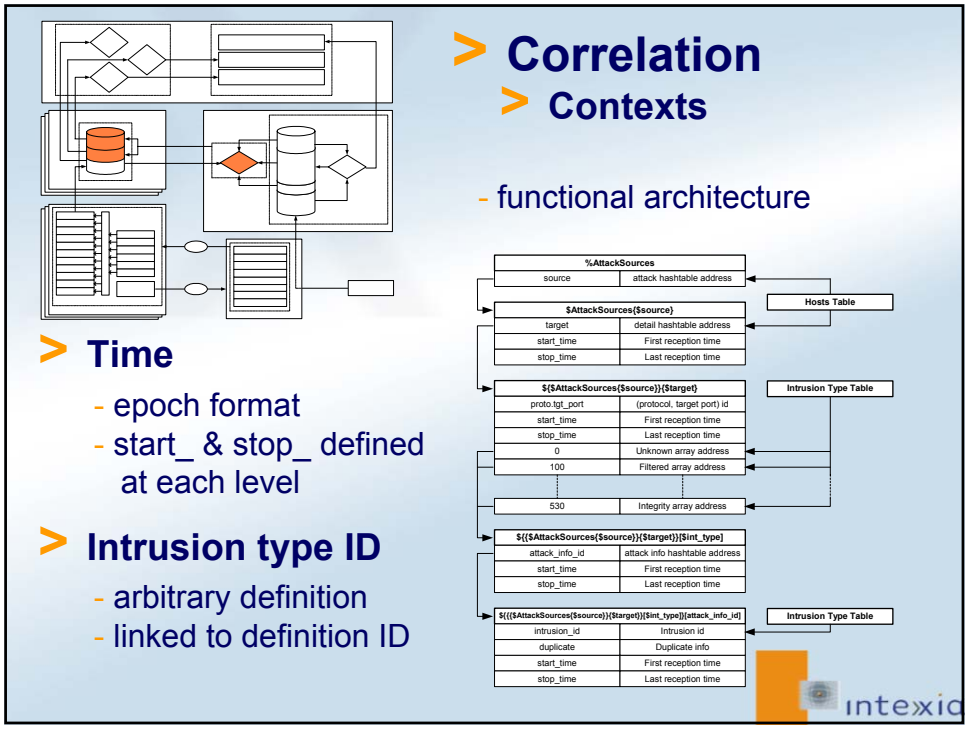
- container of formatted data matching common criteria
- multiple level of contexts may be created

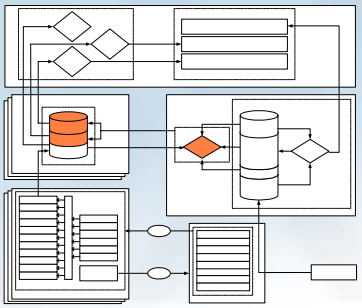
> Main context tree

- source (target) token
- target (source) token
- target proto.port
- intrusion type ID
- intrusion ID

intexio


> Correlation
> Contexts
- functional architecture
Array of source contexts
Source Host Token
Target Host Token
Target Host Token
Target Host Token
Target Host Token
Target proto.port
Target proto.port
Target proto.port
Int. Type ID
Int. Type ID
Int. Type ID
Int. Type ID
Intrusion ID
Intrusion ID
Intrusion ID
Intrusion ID
Intrusion ID
intexio

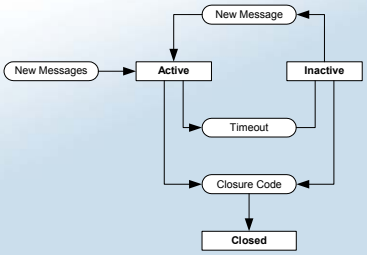

> Correlation
> Contexts
- functional architecture
%AttackSources
source
attack hashtable address
Hosts Table
$AttackSources{$source}
target
detail hashtable address
start_time
First reception time
stop_time
Last reception time
${$AttackSources{$source}}{$target}
Intrusion Type Table
proto.tgt_port
(protocol, target port) id
start_time
First reception time
stop_time
Last reception time
0
Unknown array address
100
Filtered array address
530
Integrity array address
${{$AttackSources{$source}}{$target}}{$int_type}
attack_info_id
attack info hashtable address
start_time
First reception time
stop_time
Last reception time
${{{$AttackSources{$source}}{$target}}{$int_type}}{attack_info_id}
Intrusion Type Table
intrusion_id
Intrusion id
duplicate
Duplicate info
start_time
First reception time
stop_time
Last reception time
> Time
- epoch format
- start_ & stop_ defined at each level
> Intrusion type ID
- arbitrary definition
- linked to definition ID
intexio



> Correlation

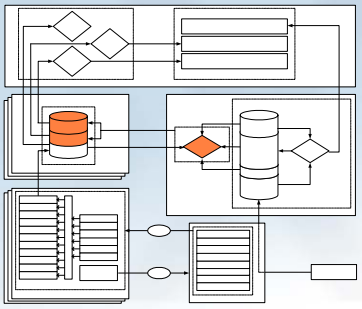
> Contexts

- context management



> Status

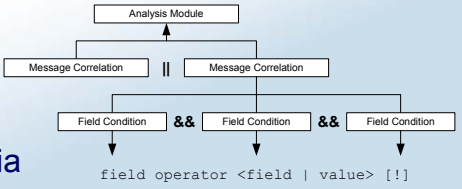
- active : on-going intrusion
- inactive : wait state
- closed : self-explanatory



> Correlation

> Structural analysis

- intrusion identification
- processes analysis

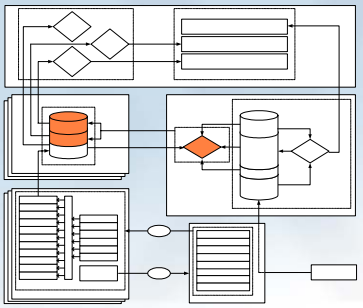


> Structure

- independant modules
- set of logical operators
- header w/ activation criteria

> Activation

- message matching header
- timer



> Correlation

> Advanced correlation

- intrusion path analysis
- security policy matching

> Functional analysis

- request to the K Box for Intrusion ID & Host Token
- criticity evaluation
- new message generation
- context closure

> Behavior analysis

- same modular process as structural analysis

intexio

> Conclusion

> Complexity of SOC setup

- integration of heterogeneous modules
- emerging standards to reduce the gap with theory

> Supervision NOW

- keep in touch with actual researches
- need for a pragmatic approach

intexio