IRSS: Incident Response Support System

Ing. Gianluca Capuzzi

Research Project Team

- Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione, Università Politecnica delle Marche
- Ing. Gianluca Capuzzi
- Ing. Egidio Cardinale
- Ing. Claudio Cilli
- Prof. Luca Spalazzi

Introduction

- Computer and network security can be improved by three kinds of tool:
  - Tools dealing with prevention
  - Tools dealing with detection
  - Tools dealing with response
- Several systems have been proposed for the first two kinds, the response is still left to the Security Manager

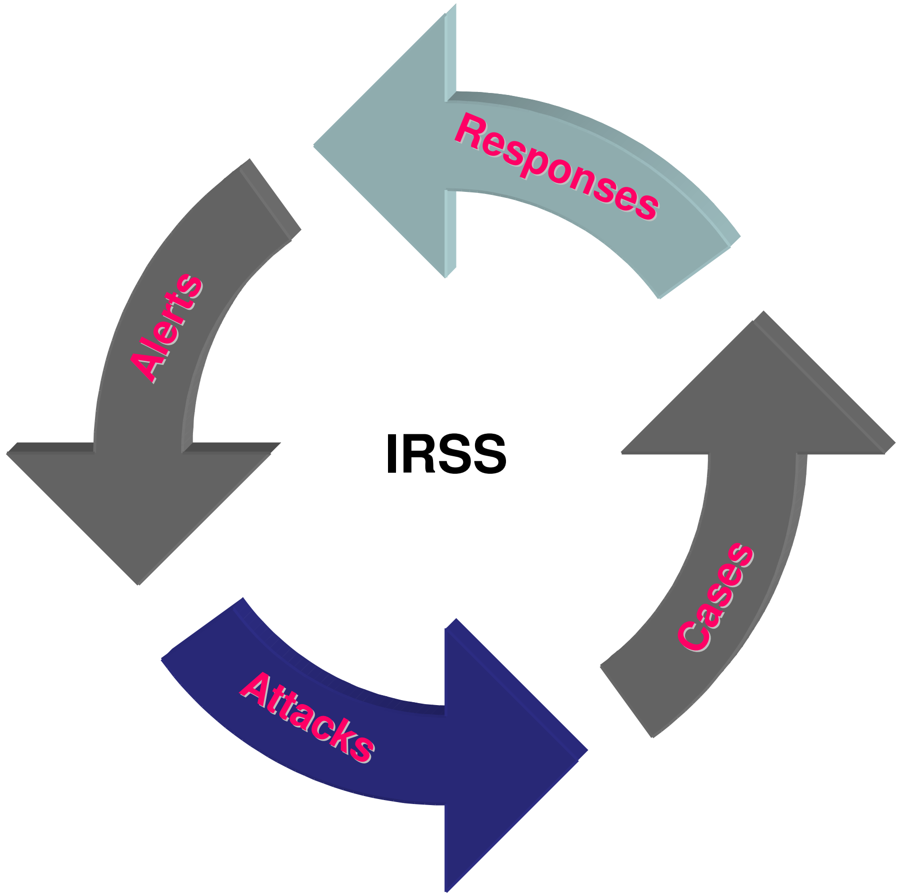
Current scenarios

- High volume of log messages (several different structures)
- Insufficiency of support systems: no integrated tools
- Timeliness of the Incident Response Activity

Aims of the work

- It creates an incident response system (we call IRSS) that supports the job of the Security Manager
- It gathers information from the other security systems (log messages)
- It correlates them to recognize attacks (set of events)
- It searches in a Knowledge Base for the closest past incident
- It returns the related plan solution

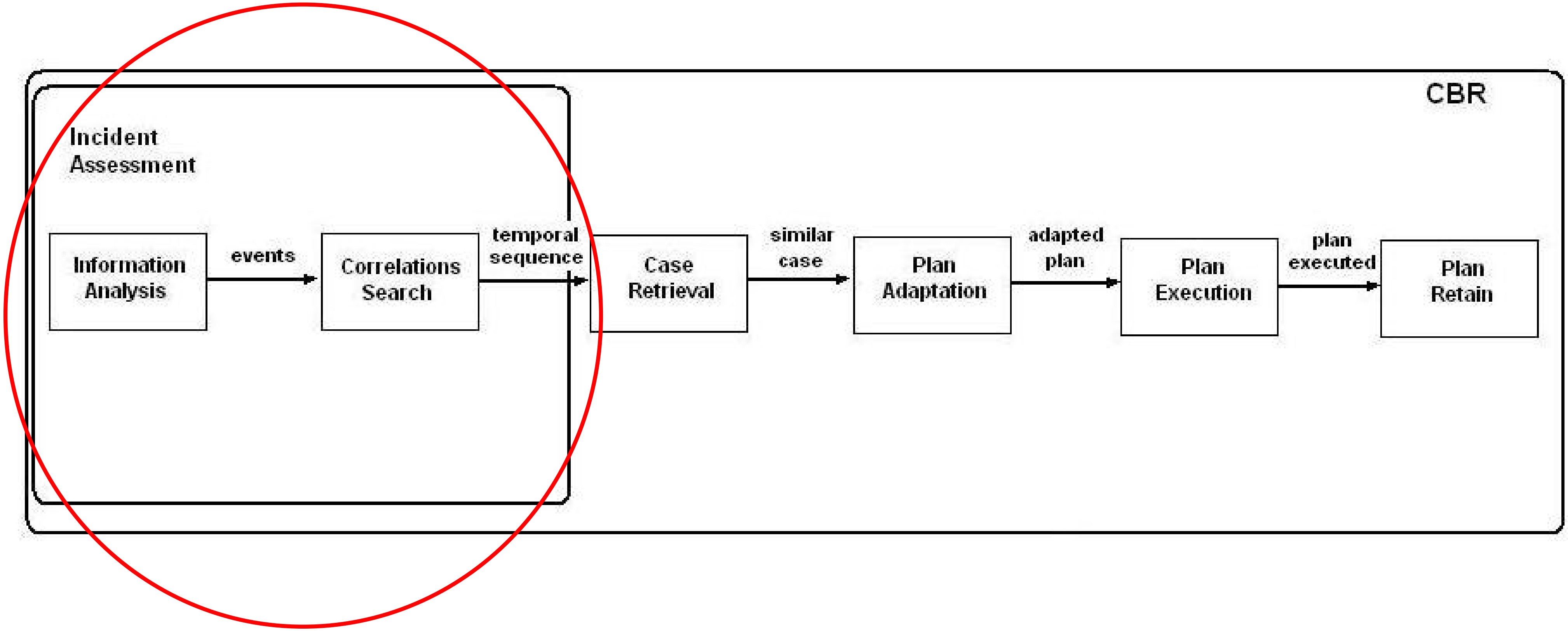
Cycle process



Solutions

- Solutions can be applied to address some problems:
  - Confortable Reading of logs
  - Further Elaboration
  - Response
  - Network Forensic Analysis

Schema of the IRSS



Schema of IRSS

- It consists of two modules:
  - A module of Incident Assessment that correlates the information in input outcoming attacks (sequences of events)
  - A Reasoner (Case-Based Reasoner) that receives the new case (attack), searches the closest case in the KB and returns the corresponding response

Related work for alert correlation

- There are many works related to this topic
- In particular, there is a paper that has a comprehensive approach to the problem of Alert Correlation: “A Comprehensive Approach to Intrusion Detection Alert Correlation”
- The authors are F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer
- This paper describes a correlation algorithm, which considers results of previous publications

Related work for CBR

- Concerning the use of Case-Based Reasoning to network security, we have only few example
- The most notable is “A Case-Based Approach to Network Intrusion Detection”
- The authors are D. G. Schwartz, S. Stoecklin and E. Yilmaz”
- This paper describes the possible application of CBR to the Intrusion Detection

About Incident Response

- Concerning Incident Response, we have:
  - Tools which deal with Intrusion Prevention working in in-line mode to block malicious connections
  - Tools dealing with Forensic Analisys
  - Tools dealing with Restore previous state (backup)
- But we have not a tool that supports the whole job of the Security Manager

Running example

- Example to explain how this system works:
  - Portscan
  - Apache exploit
  - Attempt to modify the linuxconf file

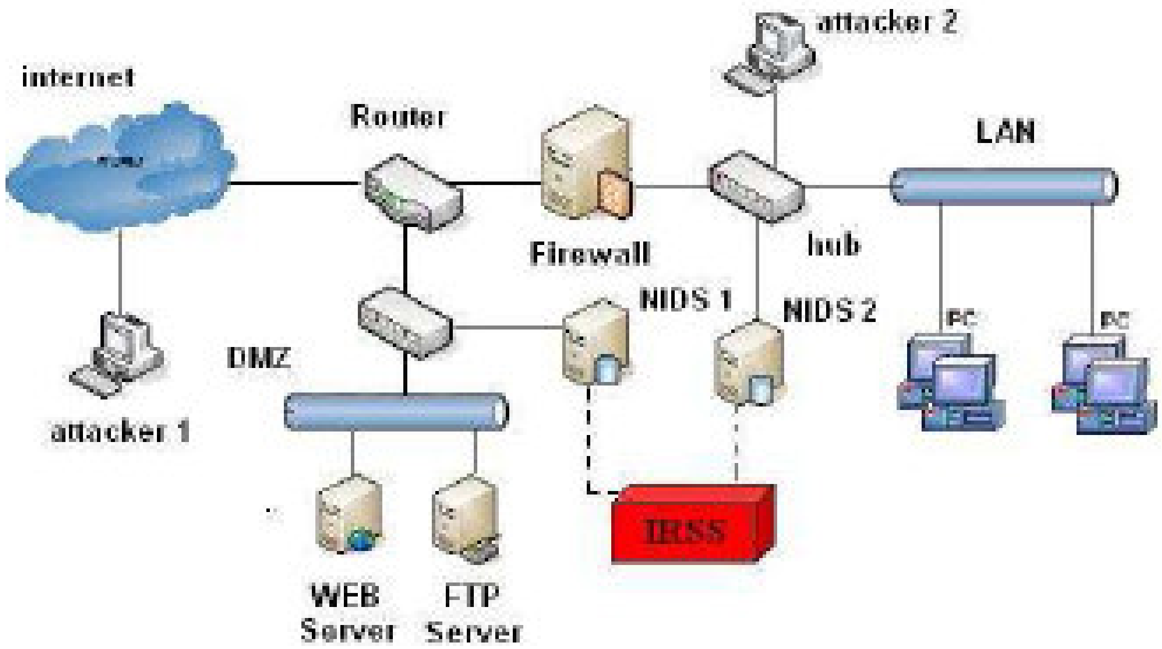
ID	Type of attack	Sensor	Start/End	Source	Target	Tag
1	IIS Exploit	N1	12.0/12.0	80.0.0.1	10.0.0.1:80	
2	Portscan	N2	10.1/14.8	31.3.3.7	10.0.0.1	
3	Portscan	N1	10.0/15.0	31.3.3.7	10.0.0.1	
4	TFTP GET passwd	N1	11.3/11.3	192.168.10.41	192.168.10.52:80	
5	TFTP GET passwd	N2	11.3/11.3	192.168.10.41	192.168.10.52:80	
6	Apache Exploit	N1	22.0/22.0	31.3.3.7	10.0.0.1:80	
7	Bad Request	A	22.1/22.1	10.0.0.1	10.0.0.1,Apache	
8	Local Exploit	H	24.6/24.6	10.0.0.1	10.0.0.1,linuxconf	
9	Local Exploit	H	24.7/24.7	10.0.0.1	10.0.0.1,linuxconf	

Running example

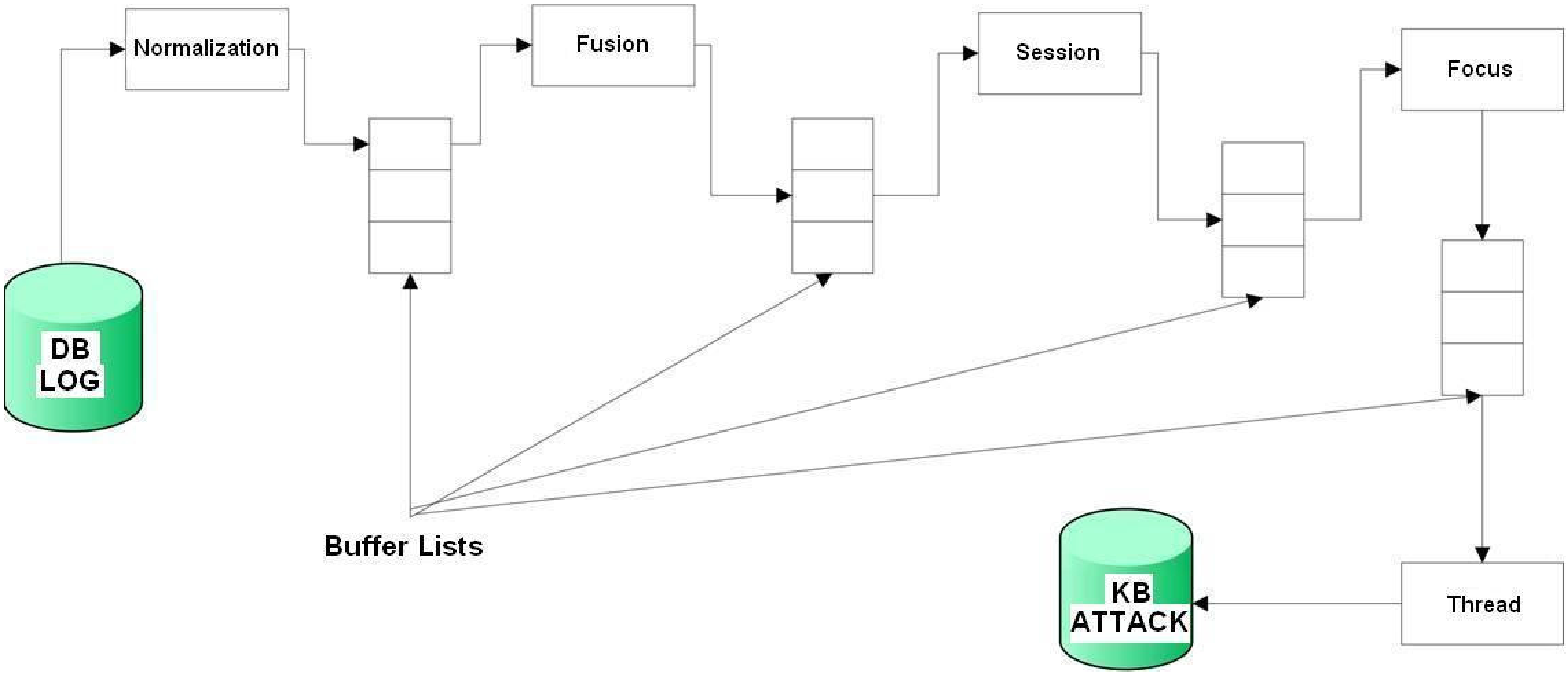
- This is an example of an intrusion consisting of three step
- We have all the log messages of the attack
- The first one is a non-relevant log event
- There are two log related to another attack

System overview

Schema of the network used to test the IRSS



Design of the Correlator



Design of the Correlator

- The Correlation is carried out in several steps
- Each step is realized by a submodule
- The input data is a set of alerts, while the output data is a set of attacks
- Each Buffer List allows transferring correlated alerts

The structure of Log message

- ID
- Message
- Sensor
- Start_time
- End_time
- Source
- Target
- Tag

Example: alert correlation

This is the result after the correlation

ID	Sensor	Start/End	Source	Target	Tag
10	N1,N2	10.0/14.8	31.3.3.7	10.0.0.1	2,3
11	N1,N2	11.3/11.3	192.168.10.41	192.168.10.52:80	4,5
12	N1,A	22.0/22.1	31.3.3.7	10.0.0.1:80	6,7
13	H	24.6/24.7	10.0.0.1	10.0.0.1,linuxconf	8,9
14	N1,N2,A,H	10.0/24.7	31.3.3.7,10.0.0.1	10.0.0.1:80,Apache,linuxconf	10,12,13
15	N1,N2	11.3/11.3	192.168.10.41	192.168.10.52:80	11

Correlation algorithm

- The correlation schema consists of four steps:
  - Fusion
  - Session reconstruction
  - Focus recognition
  - Thread reconstruction

Normalization and preprocessing

- These are not real correlation steps
- The first one normalizes log messages giving them the same structure
- The second one marks alerts non-relevant: for example, if the target is not vulnerable to this attack

Fusion

- This step aims to merge alerts produced by the same event: for istance, those produced by two sensors detecting the same packet
- It merges identical alerts whose timestamps differs no more than **Δt**
- **Δt** is the max delay of the network

Session reconstruction

- This step aims to merge alerts produced by different kinds of source
- For istance, alerts produced by Network-IDS, Host-IDS, O.S., etc.
- The resulting alert includes more information

Focus recognition

- This step aims to merge alerts produced by attacks *one-to-many* and *many-to-one*
- For istance, portscanning, DDoS attacks, etc.
- The resulting alert as one source IP and several target IPs, or viceversa

Thread reconstruction

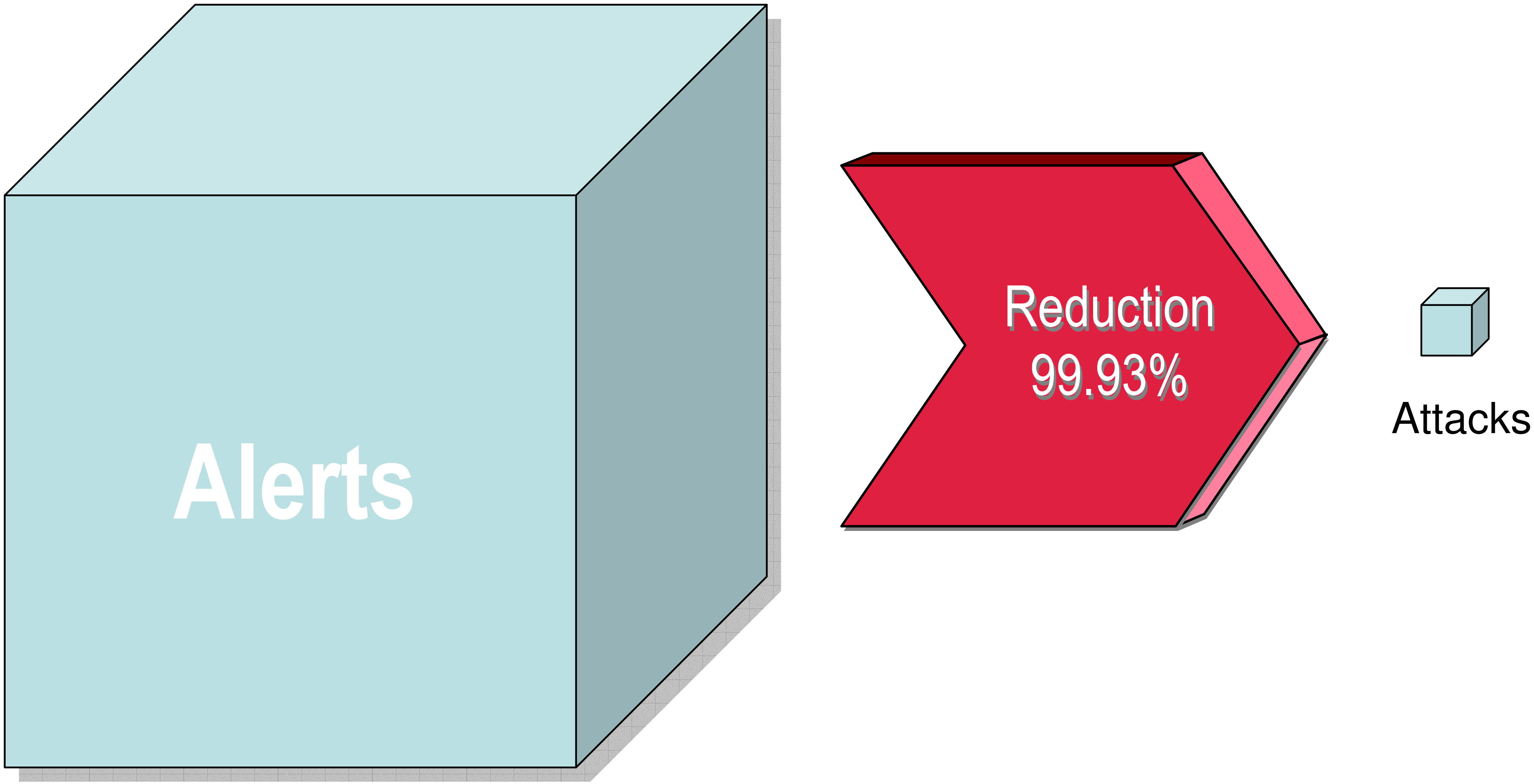
- This is the most important step
- It aims to link events related to the same attack
- It analizes alerts which have the same source IP and target
- The result is a sequence of attack steps

Experiments and results

- Two classes of experiments: DARPA Data Sets, attacks launched by ourselves
- The result of the first class:

Input Alerts	Output Alerts	Riduction Volume Alerts
45942	33	99.93%

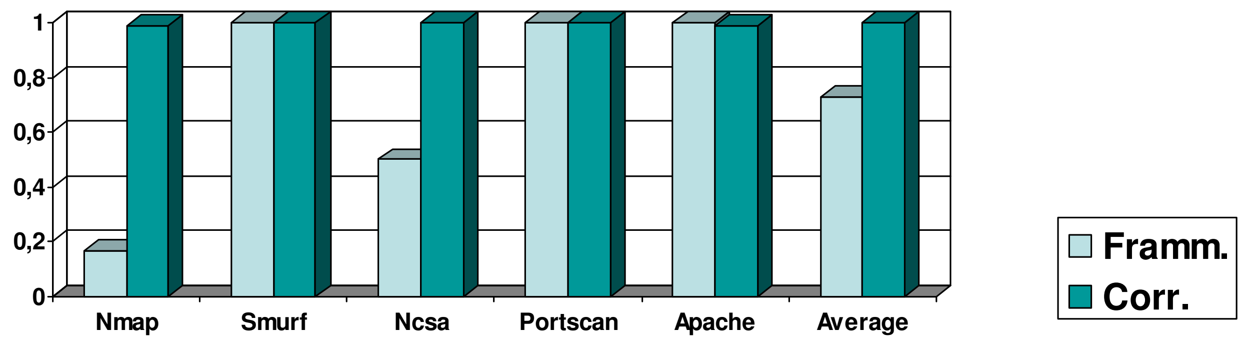
Experiments and results



Experiments and results

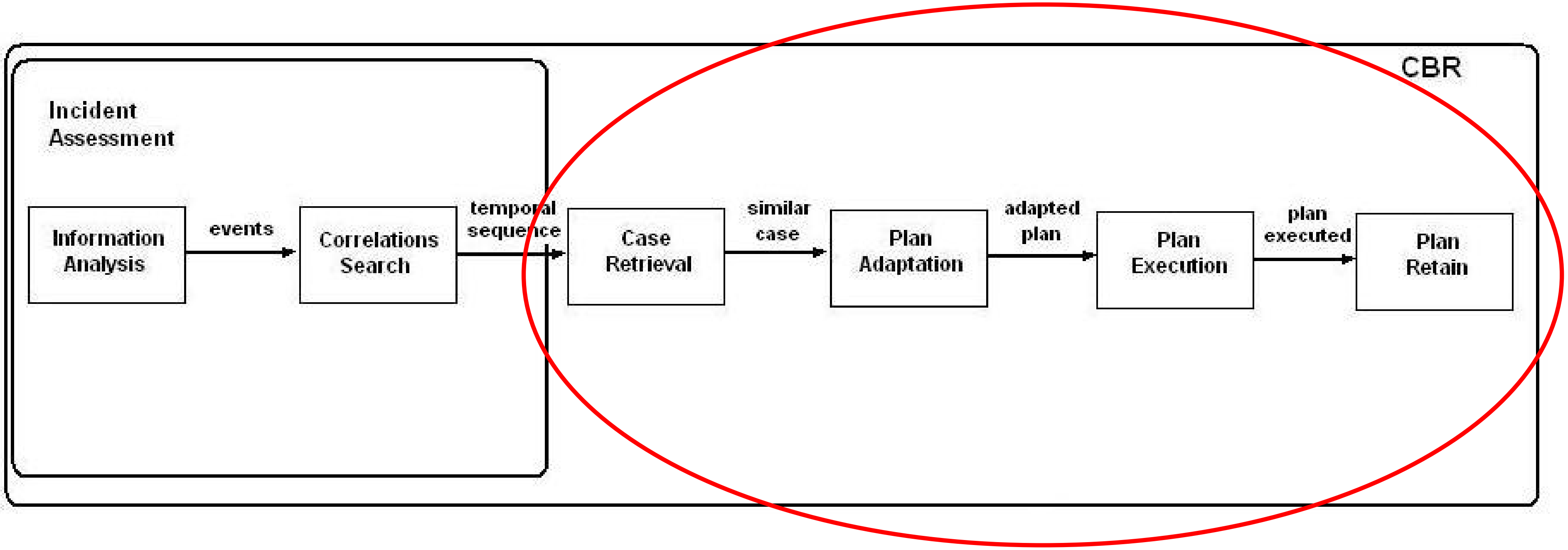
- The results of the second class:

Attack	I_{fr}	I_{corr}
nmap	0,17	0,99
smurf	1	1
ncsa	0,5	1
portscan	1	1
apache	1	0,99
Average	**0,73**	**0,998**



Design of the CBR

This is the schema of the CBR system



Incident Retrieval

- Knowledge Base:
  - Abstraction
  - Structure of Case Memory
  - Minimal Set of Cases (attacks)

	ID	Type of Attack	Sensor	Source	Target	Plan
CASE 1	C1.1	SQL Injection	N	ext	webserveraddress:httpports	PLAN 1
	C1.2	SQL Injection Basic Union	N	ext	webserveraddress:httpports	
CASE 2	C2.1	TFTP GET Passwd	N	int/ext	webserveraddress	PLAN 2
CASE 3	C3.1	Portscan	N	int/ext	webserveraddress	PLAN 3
	C3.2	Apache Exploit, Bad Request	N,A	int/ext	webserveraddress:httpports,Apache	
	C3.3	Local Exploit	H	int/ext	webserveraddress:linuxconf	
CASE 4	C4.1	Local Exploit	H	int/ext	webserveraddress	PLAN 4
	C4.2	IIS Exploit	N	int/ext	webserveraddress:httpports	
	C4.3	Portscan	N	int/ext	webserveraddress	

Incident Similarity Functions:

$$F_1(I, I_h) = ||S||$$

$$F_2(I, I_h) = \begin{cases} ||S|| & Abs(I) \subseteq Abs(I_h) \\ 0 & \text{otherwise} \end{cases}$$

$$F_3(I, I_h) = \begin{cases} ||S|| & \forall e_i, e_j \in I \\ & \forall e_l, e_k \in I_h \\ & \text{such that } Abs(e_i) = Abs(e_l) \in S \\ & \text{and } Abs(e_j) = Abs(e_k) \in S \\ & \text{then} T(e_i) < T(e_j) \text{ iff } T(e_l) < T(e_k) \\ 0 & \text{otherwise} \end{cases}$$

$$F_4(I, I_h) = \begin{cases} ||F_3(I, I_h)|| & Abs(I) \subseteq Abs(I_h) \\ 0 & \text{otherwise} \end{cases}$$

Example: incident retrieval

- For our experiments we used the Similarity Function 3
- The result is the Case 3

- An advantage of this kind of Reasoner is the possibility of manage new attacks
- Therefore, it searches for past attacks, returning the closest Case

Plan Adaptation

- The adaptation activity is devoted to the Security Manager, who has the final decision
- Now, we follow the basic kind of adaptation, by replacing the abstract attributes with their current values
- After that, it is submitted for validation to the Security Manager

Example of plan

- A plan consists of commands and messages
- In the example we use the Linux commands for:
  - Sending a message to Security Manager
  - Updating Systems (O.S., Security Systems, etc.)
  - Removing malicious code
  - Communicating to the operator

1) mail SecurityAdmin
2) apt-get update Systems && apt-get upgrade Systems | echo *"control update actions and configuration of security systems"*
3) /usr/bin/removaltool
4) echo *"restore previous status using backup copy and update it"*

ISACA® Servire i professionisti dell'IT Governance Rome Chapter

Execution

ISACA® Serving IT Governance Professionals

- The output report is presented to the Security Manager who can change and execute it
- During the execution stage the Security Manager has the possibility of evaluating the effectiveness of the response plan
- He can correct the plan or add other instructions to complete it

Retainment

- After the execution process, the plan has been corrected and evaluated
- Hence, the Security Manager can update the Knowledge Base (Case Memory)
- This is an important aspect of the CBR, which is able to learn new cases
- In this way, it can help to solve the problem of managing new attacks

Conclusions

- The IRSS provides a whole picture about what the Security Manager has to do, coordinating all activities (it's original)
- It learns new cases with their responses (manage new attacks)
- We have implemented a prototype: first results
- We have planned:
  - to investigate new similarity metrics and more sophisticated adaptation algorithms
  - to perform more experiments