ISACA
Servire i professionisti dell'IT Governance
Rome Chapter

IRSS: Incident Response Support System: nuovi sviluppi 29 marzo 2007

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. Ivan Di Pietro
- 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

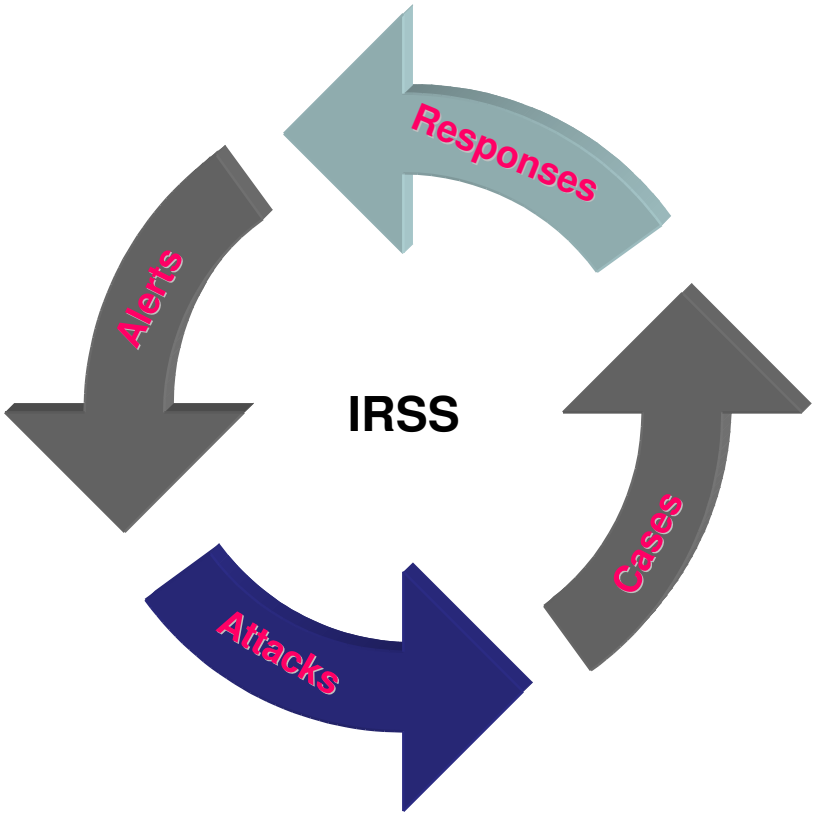
Current scenario

- 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

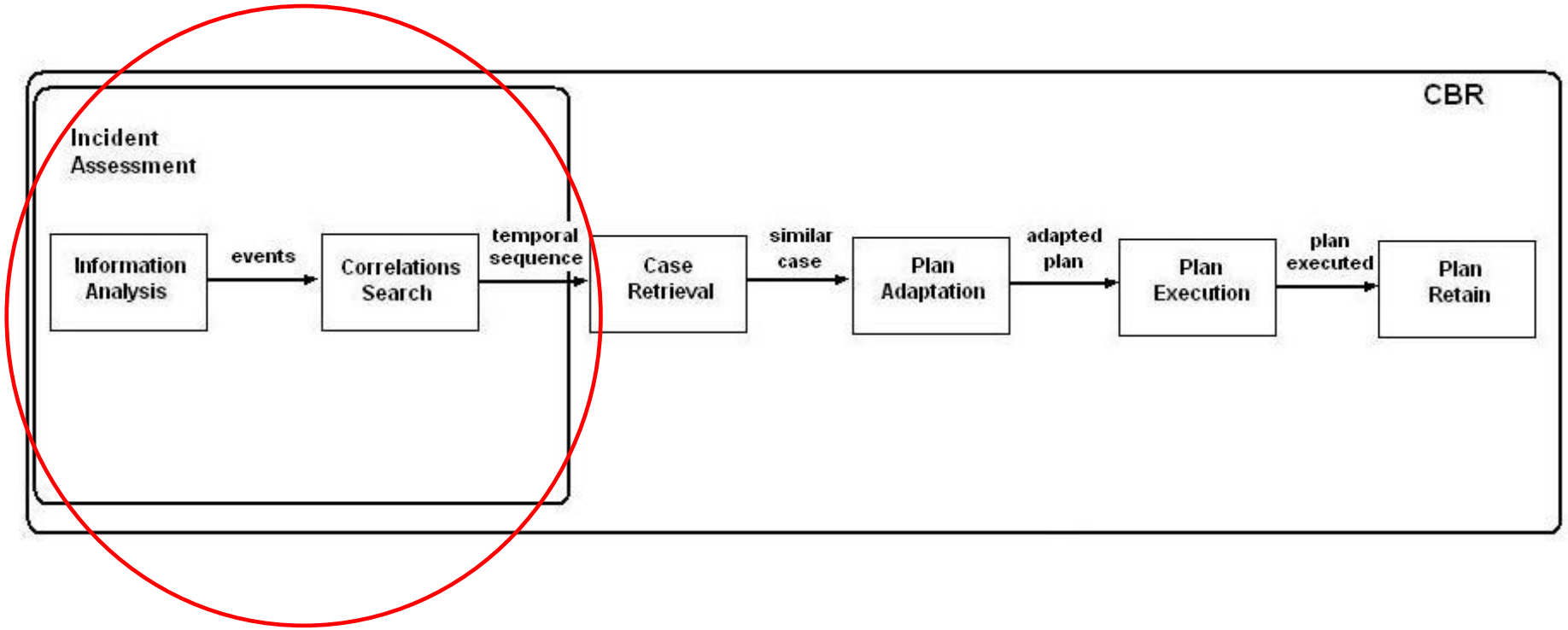
Cycle process



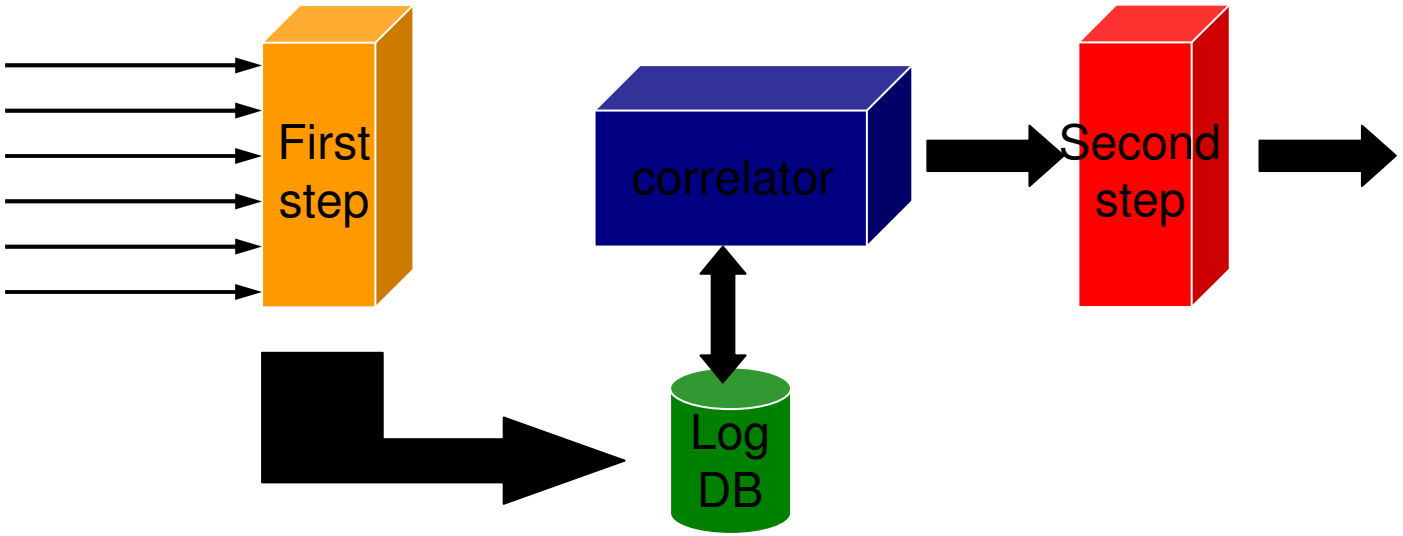
Solutions

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

Schema of the IRSS



Filters



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 Analysis
  - 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

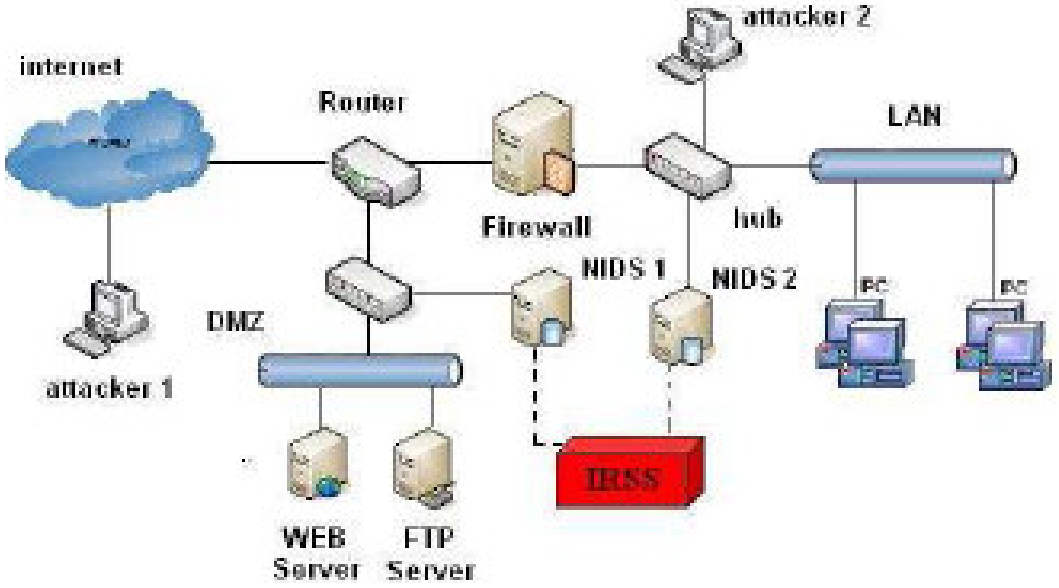
ID	Type of attack	Sensor	Start/End	Source	Target	Tag
1	IIS Exploit	N1	12.0/12.0	80.0.0.1	10.0.0.1:80	
2	Portscan	N2	10.1/14.8	31.3.3.7	10.0.0.1	
3	Portscan	N1	10.0/15.0	31.3.3.7	10.0.0.1	
4	TFTP GET passwd	N1	11.3/11.3	192.168.10.41	192.168.10.52:80	
5	TFTP GET passwd	N2	11.3/11.3	192.168.10.41	192.168.10.52:80	
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 steps
- 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

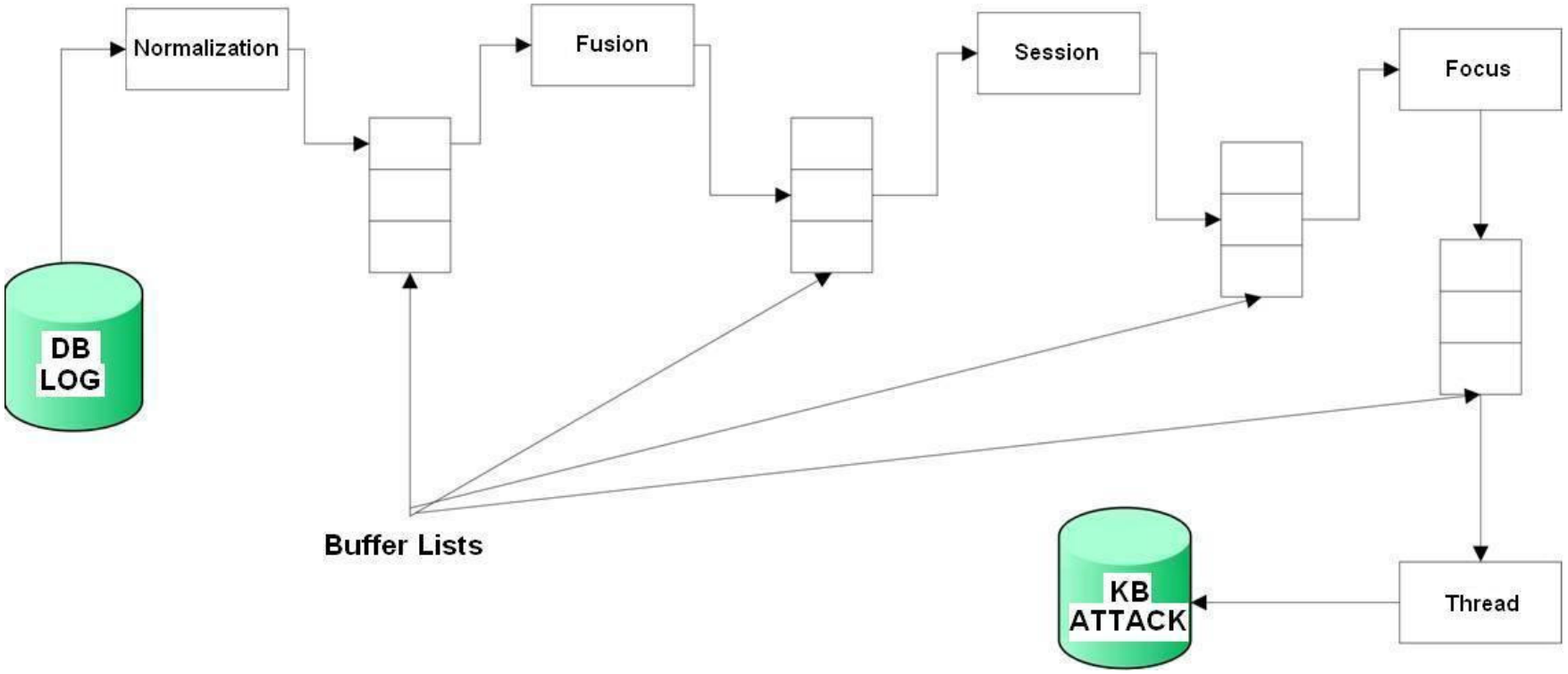
Schema of the network used to test the IRSS



Filter: 1st step

- It consists of a white lists
- They are lists of rules to filter log files in order to gather only log messages related to security activities
- One list for each log file
- Example: in secure file of Linux OS, it does not take messages about the starting and the stopping of the services (sshd, httpd, etc.)

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 the 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	10.0/14.6	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 a 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

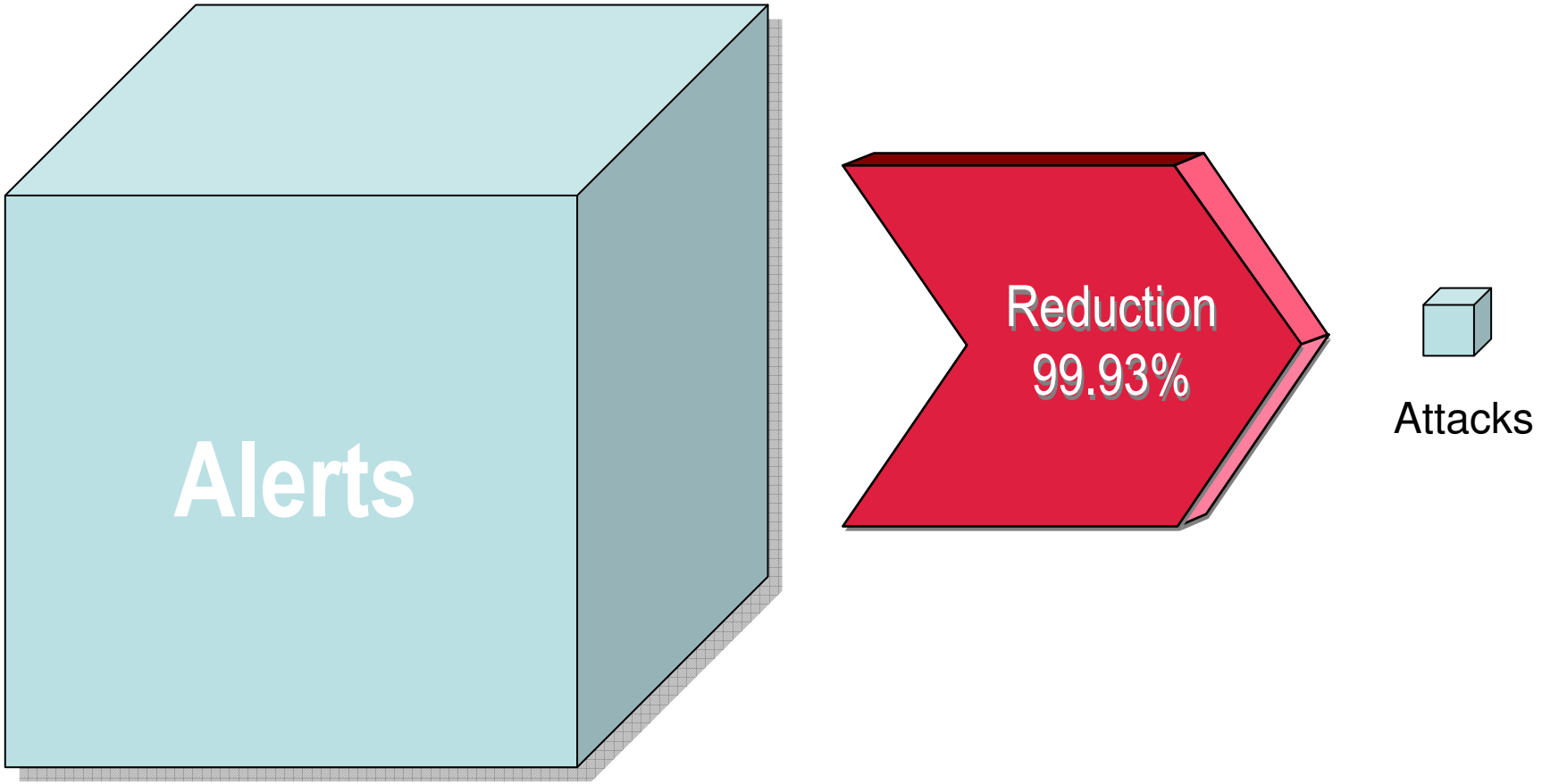
- This is the most important step
- It aims to link events related to the same attack
- It analizes alerts which have the same source IP and target
- The result is a sequence of attack steps

Experiments and results

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

Input Alerts	Output Alerts	Reduction Volume Alerts
45942	33	99.93%

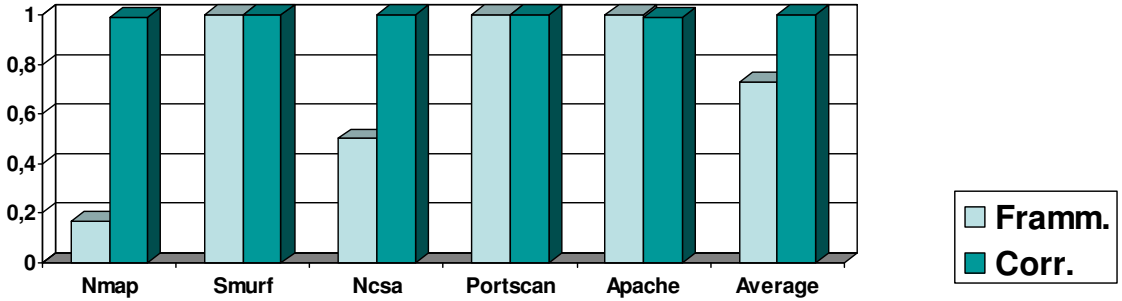
Experiments and results



Experiments and results

- The result 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**



Filter: 2nd step

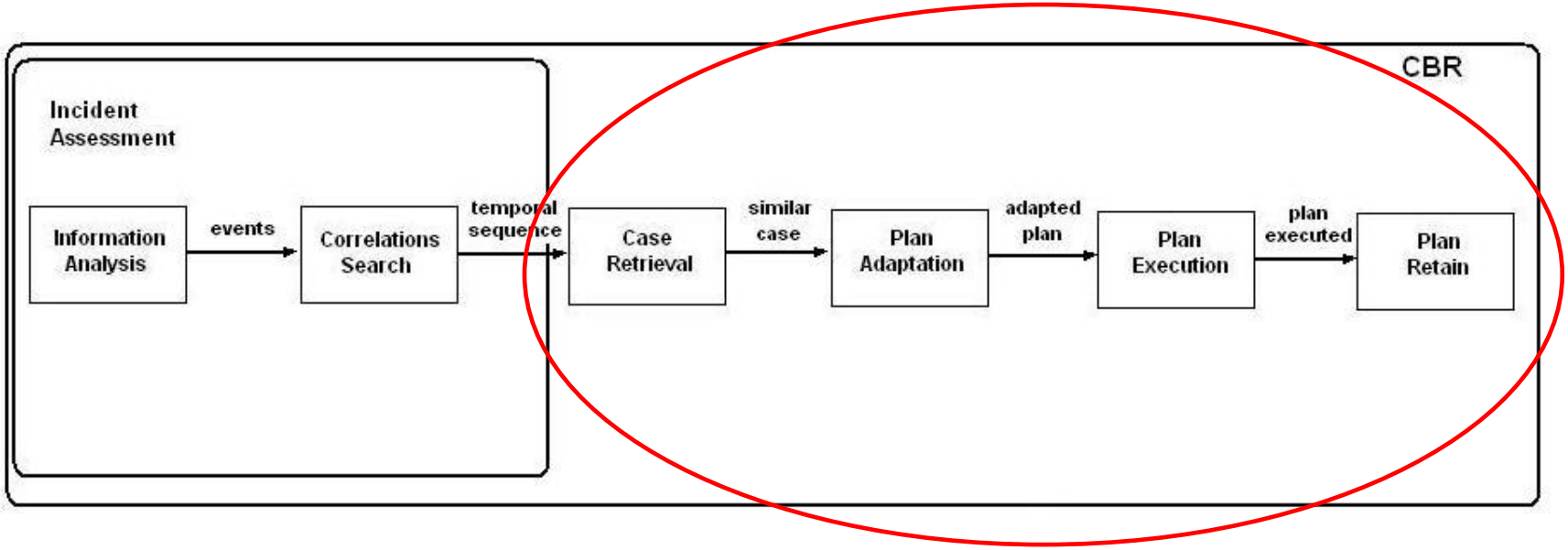
- Entropy-based filter
- Weight associated to each alert: $w_t = -\log_2 p(t)$
- Where ***p(t) = 1/f(t)***
- and ***f(t)*** is the frequency of the alert ***t***
- In the log DB, each alert has its entropy weight w_t

Experimental results

	Ratios	Adjusted LT	Adjusted HT	Trivial attacks %	FP	FN
4w_mon	[20,23 - 10,52]	0,450	1,050	62,35	240	0
	[14,5 – 15]	0,800	1,100	91,81	44	2
4w_tue	[20,23 - 10,52]	0,500	1,250	69,52	252	0
	[14,5 – 15]	0,800	1,100	92,85	54	1
4w_wed	[20,23 - 10,52]	0,450	1,050	74,04	360	0
	[14,5 – 15]	0,800	1,050	93,63	81	1
4w_thu	[20,23 - 10,52]	0,450	1,050	69,75	470	0
	[14,5 – 15]	0,700	1,050	87,13	196	0
4w_fri	[20,23 - 10,52]	0,450	1,150	70,48	364	0
	[14,5 – 15]	0,750	1,150	91,96	94	0
5w_mon	[20,23 - 10,52]	0,450	1,050	62,15	433	1
	[14,5 – 15]	0,700	1,100	84,67	167	1
5w_tue	[20,23 - 10,52]	0,500	1,250	72,64	342	1
	[14,5 – 15]	0,800	1,050	93,85	68	3
5w_wed	[20,23 - 10,52]	0,450	1,150	71,78	384	1
	[14,5 – 15]	0,750	1,050	93,02	89	1
5w_thu	[20,23 - 10,52]	0,450	1,900	47,61	1201	1
	[14,5 – 15]	0,650	1,100	63,87	826	4
5w_fri	[20,23 - 10,52]	0,450	1,050	70,47	387	0
	[14,5 – 15]	0,750	0,950	88,08	149	0

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	$C_{1.1}$	SQL Injection	N	ext	webserveraddress:httpports	PLAN 1
	$C_{1.2}$	SQL Injection Basic Union	N	ext	webserveraddress:httpports	
CASE 2	$C_{2.1}$	TFTP GET Passwd	N	int/ext	webserveraddress	PLAN 2
CASE 3	$C_{3.1}$	Portscan	N	int/ext	webserveraddress	PLAN 3
	$C_{3.2}$	Apache Exploit, Bad Request	N,A	int/ext	webserveraddress:httpports,Apache	
	$C_{3.3}$	Local Exploit	H	int/ext	webserveraddress:linuxconf	
CASE 4	$C_{4.1}$	Local Exploit	H	int/ext	webserveraddress	PLAN 4
	$C_{4.2}$	IIS Exploit	N	int/ext	webserveraddress:httpports	
	$C_{4.3}$	Portscan	N	int/ext	webserveraddress	

Incident Retrieval

Incident Similarity Functions:

$$F_1(I_c, I_k) = \frac{\sum_{\forall t \in Type(I_c)} \min(n_t^{(c)}, n_t^{(k)})}{\sum_{\forall t \in Type(I_c)} n_t^{(c)}}$$

$$F_2(I_c, I_k) = \frac{\sum_{\forall t \in Type(I_c)} \min(m_t^{(c)}, m_t^{(k)})}{\sum_{\forall t \in Type(I_c)} m_t^{(c)}}$$

$$F_3(I_c, I_k) = \frac{\sum_{\forall t \in Type(I_c)} w_t \cdot \min(n_t^{(c)}, n_t^{(k)})}{\sum_{\forall t \in Type(I_c)} w_t \cdot n_t^{(c)}}$$

$$F_4(I_c, I_k) = \frac{\sum_{\forall t \in Type(I_c)} w_t \cdot \min(m_t^{(c)}, m_t^{(k)})}{\sum_{\forall t \in Type(I_c)} w_t \cdot m_t^{(c)}}$$

Experimental results

Similarity	Threshold	Experiment 1			Experiment 2		
		Recall	Precision	Time	Recall	Precision	Time
F_1	0.7	54.72%	93.55%	49sec	54.14%	80.00%	39sec
F_2	0.7	77.36%	95.35%	44sec	73.81%	81.58%	26sec
F_3	0.7	56.60%	93.75%	111sec	60.71%	69.79%	32sec
F_4	0.7	81.13%	93.48%	95sec	84.52%	83.53%	22sec
"	0.6	84.90%	91.84%		86.90%	82.95%	22sec

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

Abstract Actions	Concrete Actions	Preconditions
check the process list	taskmgr.exe ps -aux >> proclist.txt	OS=Microsoft Windows OS=Linux
compare the process list to the reliable one	diff proclist.txt reliabprocs.txt >> badprocs.txt winmerge.exe proclist.txt reliabprocs.txt >> badprocs.txt	OS=Linux OS= Microsoft Windows
check the connection list	netstat >> ports.txt	OS=Microsoft Windows, Linux
compare the connection list to the reliable one	diff network.txt reliabnet.txt >> badconn.txt fc network.txt reliabnet.txt >> badconn.txt	OS=Microsoft Windows, Linux OS=Microsoft Windows
check the register list and compare it to the reliable list	diff regedit.txt reliabregedit>> modifiedKeys.txt	OS=Microsoft Windows
kill illegal processes	badprocs.txt >> kill -9 End Process	OS=Linux OS=Microsoft Windows
remove vulnerabilities installing patches, updating systems remove modified keys listed in modifiedkeys.txt	update windows yum update update explorer update firefox update antivirus	internet connection or update-CD/DVD Fedora Linux with Internet connection MS IExplorer Mozilla Firefox antivirus present
configure the firewall to reject traffic from the exploited ports	ports.txt >>access-list 101 deny tcp eq 25 ports.txt >> iptables -p –dport -j REJECT,	CISCO firewall iptables, Linux
restore the normal status of the network	use backup copies to restore services use backup copies to restore corrupted softwares use backup copies to restore damaged files	backup and backup utility backup and backup utility backup and backup utility

Result

Response plan
ps -aux >> proclist.txt diff proclist.txt reliabprocs.txt >> badprocs.txt lsof >> ports.txt diff ports.txt reliabports.txt >> badports.txt
badprocs.txt >> kill -9 yum update update firefox update antivirus badports.txt >> iptables -p –dport -j REJECT
use backup copies to restore services use backup copies to restore corrupted softwares use backup copies to restore damaged files

Execution

- 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