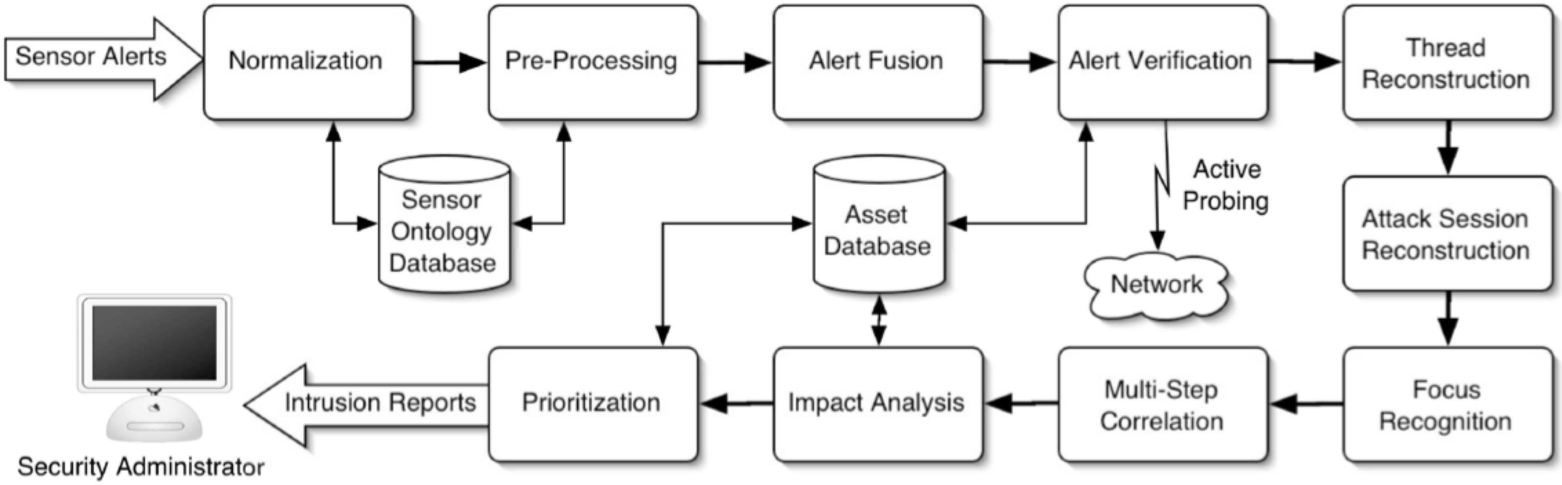
CS259D: Data Mining for Cybersecurity

Alert correlation

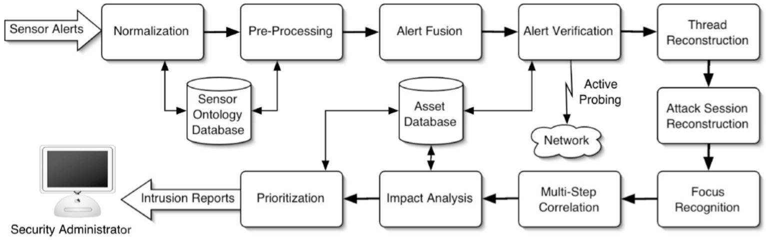
- Different attack manifestations
  - Network packets
  - OS calls
  - Audit records
  - Application logs
- Different types of intrusion detection
  - Host vs network
  - IT environment (e.g., Windows vs Linux)
  - Levels of abstraction (e.g., Kernel level vs application level)
- Goal:
  - Aggregate outputs of multiple IDSs
  - Filter out irrelevant alerts
  - Provide succinct view of security-related activity on the network

Architecture



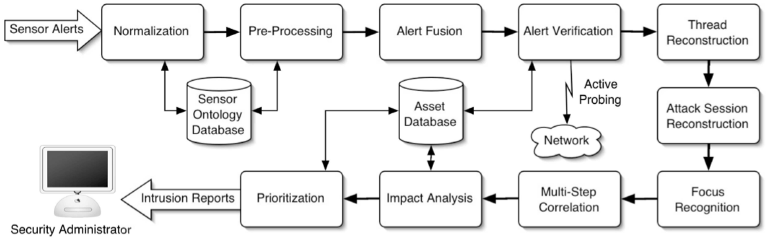
Components

- **Normalization:** translate alerts to a common format
- **Preprocessing:** augment normalized alerts by assigning meaningful values to all alert attributes
    - Start time, end time
    - Source, target



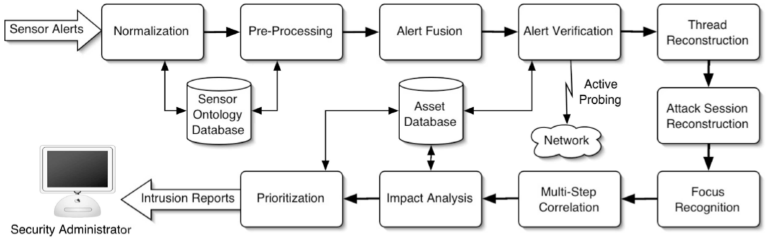
Components

- **Fusion:** combine alerts representing the same attack by different IDSs
- **Verification:** determine the success of the attack corresponding to the alert
- **Thread reconstruction:** combine series of alerts due to attacks by a single attacker against a single target
- **Session reconstruction:** associate network-based alerts and host-based alerts



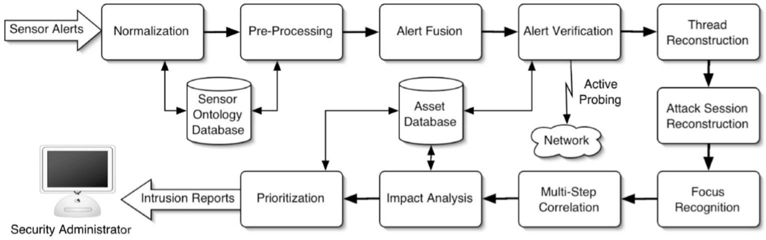
Components

- **Focus recognition:** identify hosts that are source or target of many attacks
  - DoS, port scanning
- **Multistep correlation:** identify common attack patterns
  - Sequence of individual attacks at different points of network
  - Example: Island hopping



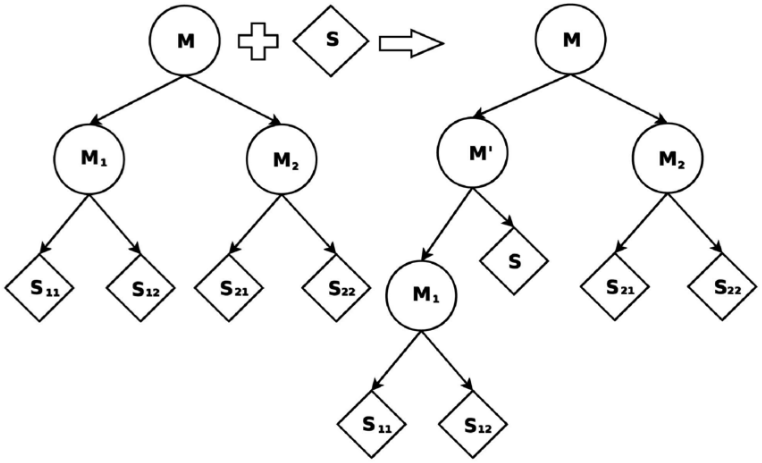
Components

- **Impact analysis:** determine the attack impact for the specific network
- **Prioritization:** Assign priorities to alerts



Meta-alerts

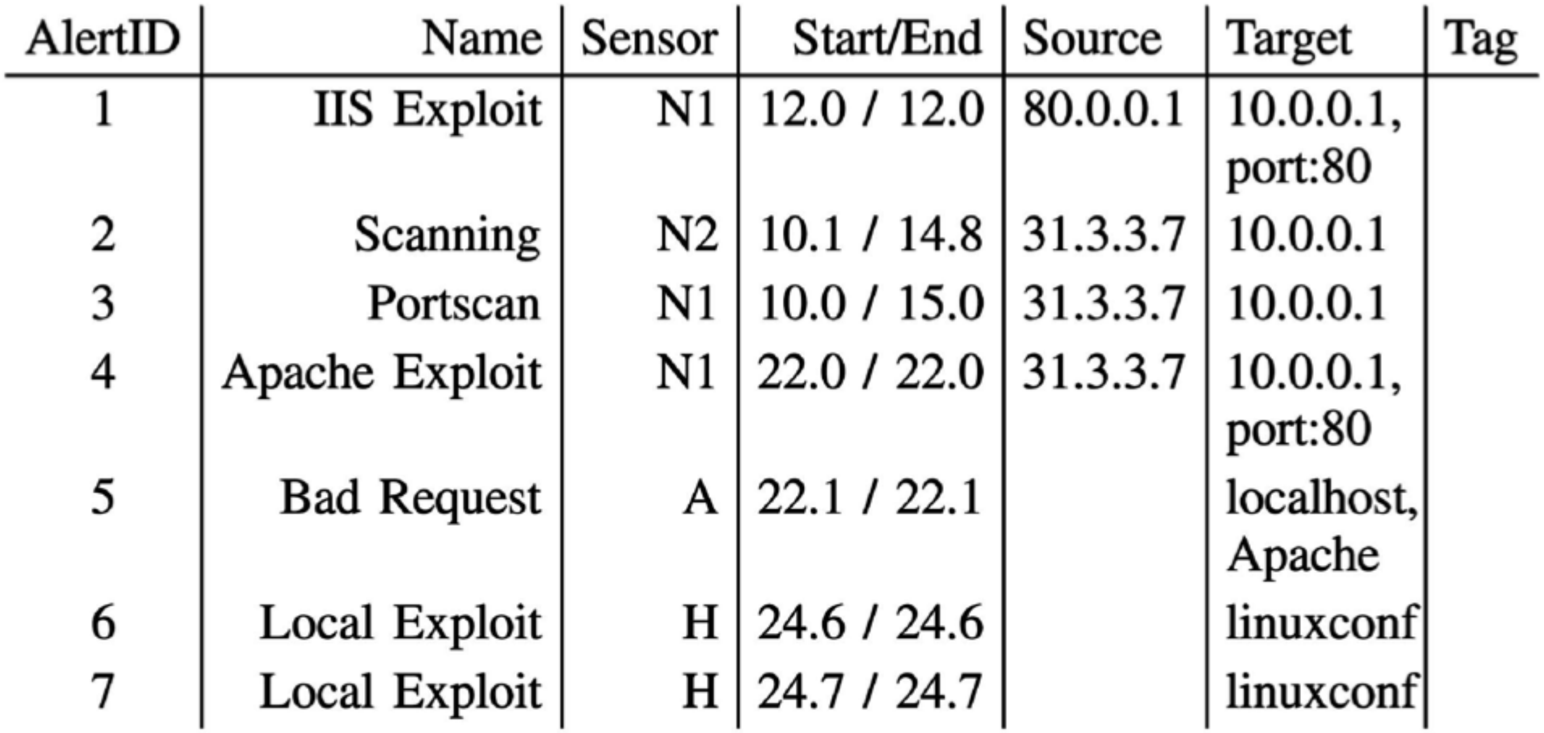
- Definition:
  - Higher-level alerts made via merging
  - Attribute values derived from those of original alerts
- Example:
  - a "portscan" alert composed of a series of alerts referring to individual network probe packets
  - Target attribute: all hosts that were port-scanned
- Representation:
  - A tree with IDS alerts at the leaves
  - Merging done in a BFS fashion



Example attack scenario

- Vulnerable Apache Web service on a Linux host (IP: 10.0.0.1)
- Host-based IDS (H)
- Application-based IDS (A): monitors Apache Web logs for malicious activity
- Two different network-based IDSs (N1 and N2)

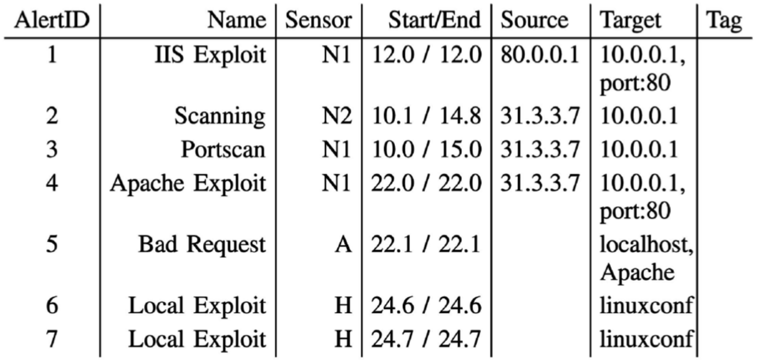
Example attack scenario



AlertID	Name	Sensor	Start/End	Source	Target	Tag
1	IIS Exploit	N1	12.0 / 12.0	80.0.0.1	10.0.0.1, port:80	
2	Scanning	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	Apache Exploit	N1	22.0 / 22.0	31.3.3.7	10.0.0.1, port:80	
5	Bad Request	A	22.1 / 22.1		localhost, Apache	
6	Local Exploit	H	24.6 / 24.6		linuxconf	
7	Local Exploit	H	24.7 / 24.7		linuxconf	

Example attack scenario

- Attacker (IP: 31.3.3.7) first portscans host
  - Discovers vulnerable Apache server (Alerts 2, 3)
- During scan a worm (IP: 80.0.0.1) attempts Microsoft IIS exploit and fails (Alert 1)
- After scan, attacker exploits Apache buffer overflow (Alerts 4, 5)
  - Gets interactive shell as apache user
- Using a local exploit against linuxconf, attacker becomes root (Alerts 6, 7)



AlertID	Name	Sensor	Start/End	Source	Target	Tag
1	IIS Exploit	N1	12.0 / 12.0	80.0.0.1	10.0.0.1, port:80	
2	Scanning	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	Apache Exploit	N1	22.0 / 22.0	31.3.3.7	10.0.0.1, port:80	
5	Bad Request	A	22.1 / 22.1		localhost, Apache	
6	Local Exploit	H	24.6 / 24.6		linuxconf	
7	Local Exploit	H	24.7 / 24.7		linuxconf	

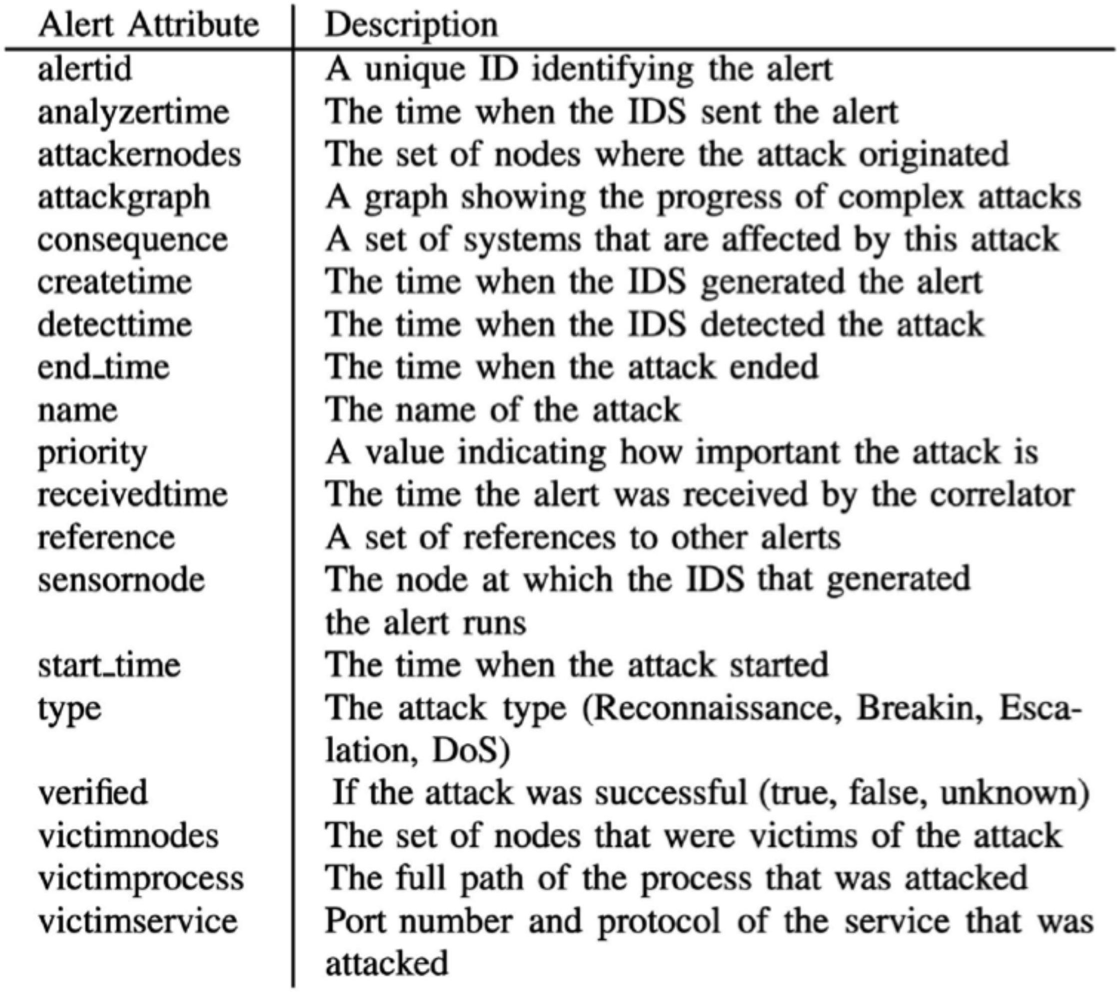
Example attack scenario

- Desired output of correlation: Single meta-alert for a multi-step attack against victim host
  - Step 1: Initial scanning (Alerts 2, 3)
  - Step 2: Remote attack against web server (Alerts 4, 5)
  - Step 3: Privilege escalation (Alerts 6, 7)
- Alert 1 should be discarded as irrelevant

Alert normalization

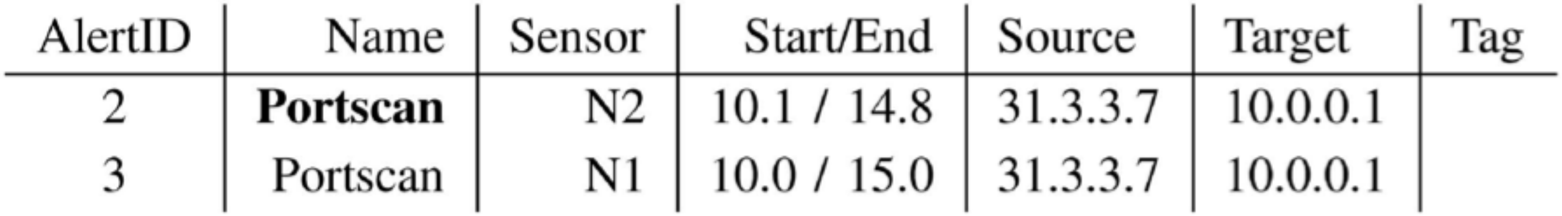
- Unify alert formats
- Example: Intrusion Detection Message Exchange Format (IDMEF)
  - Proposed by the Internet Engineering Task Force
- Implemented using wrapper modules for different IDSs

Alert normalization



Alert Attribute	Description
alertid	A unique ID identifying the alert
analyzertime	The time when the IDS sent the alert
attackernodes	The set of nodes where the attack originated
attackgraph	A graph showing the progress of complex attacks
consequence	A set of systems that are affected by this attack
createtime	The time when the IDS generated the alert
detecttime	The time when the IDS detected the attack
end_time	The time when the attack ended
name	The name of the attack
priority	A value indicating how important the attack is
receivedtime	The time the alert was received by the correlator
reference	A set of references to other alerts
sensornode	The node at which the IDS that generated the alert runs
start_time	The time when the attack started
type	The attack type (Reconnaissance, Breakin, Escalation, DoS)
verified	If the attack was successful (true, false, unknown)
victimnodes	The set of nodes that were victims of the attack
victimprocess	The full path of the process that was attacked
victimservice	Port number and protocol of the service that was attacked

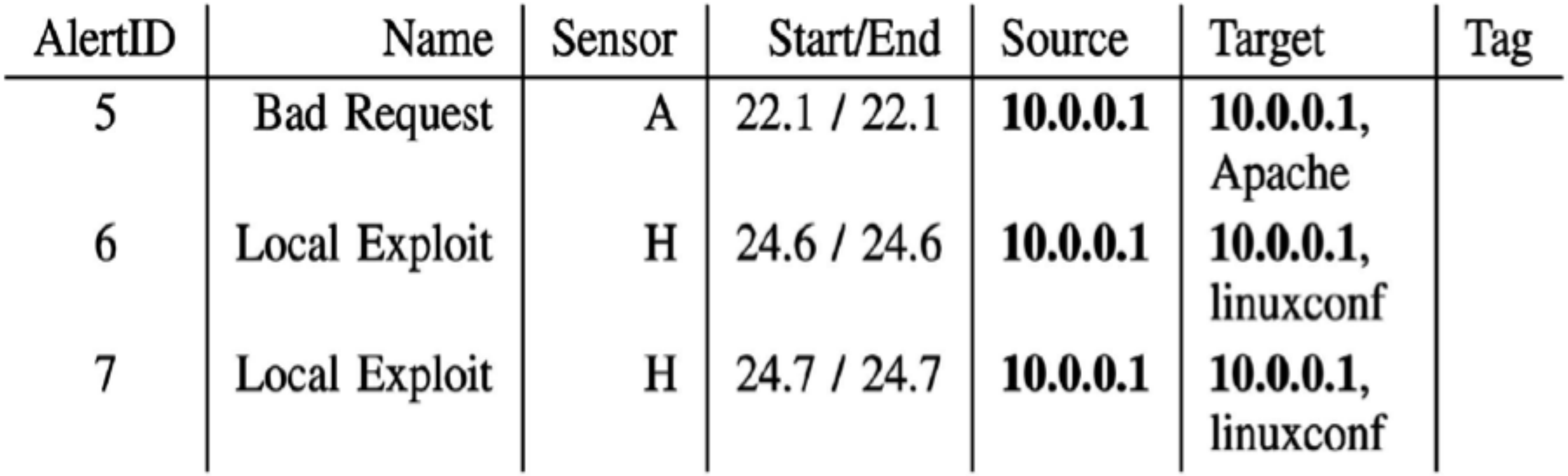
Alert normalization



AlertID	Name	Sensor	Start/End	Source	Target	Tag
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	

Alert preprocessing

- Supply missing alert attributes as accurately as possible
  - Use several heuristics

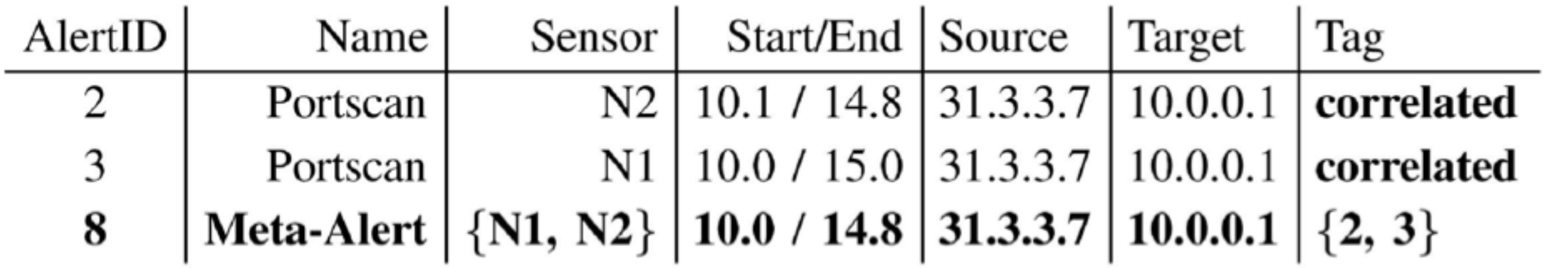


AlertID	Name	Sensor	Start/End	Source	Target	Tag
5	Bad Request	A	22.1 / 22.1	**10.0.0.1**	**10.0.0.1,** Apache	
6	Local Exploit	H	24.6 / 24.6	**10.0.0.1**	**10.0.0.1,** linuxconf	
7	Local Exploit	H	24.7 / 24.7	**10.0.0.1**	**10.0.0.1,** linuxconf	

Alert fusion

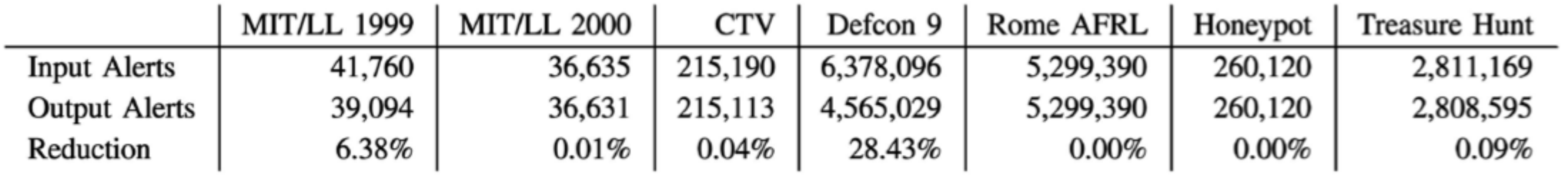
- Goal: Combine alerts representing independent detection of a same attack by different IDSs
- Fusion: Temporal difference between alerts and information they contain
  - Keep sliding time window of alerts
  - Alerts within the time window stored in a time-ordered queue
  - Upon new alert, compared to alerts in queue
  - Match if all overlapping attributes are equal and new alert is produced by a different sensor
  - Upon a match, alerts are merged; resulting meta-alert replaces the matched alert in the queue

Alert fusion



AlertID	Name	Sensor	Start/End	Source	Target	Tag
2	Portscan	N2	10.1 / 14.8	31.3.3.7	10.0.0.1	**correlated**
3	Portscan	N1	10.0 / 15.0	31.3.3.7	10.0.0.1	**correlated**
8	**Meta-Alert**	{**N1, N2**}	**10.0 / 14.8**	**31.3.3.7**	**10.0.0.1**	{**2, 3**}

Alert fusion

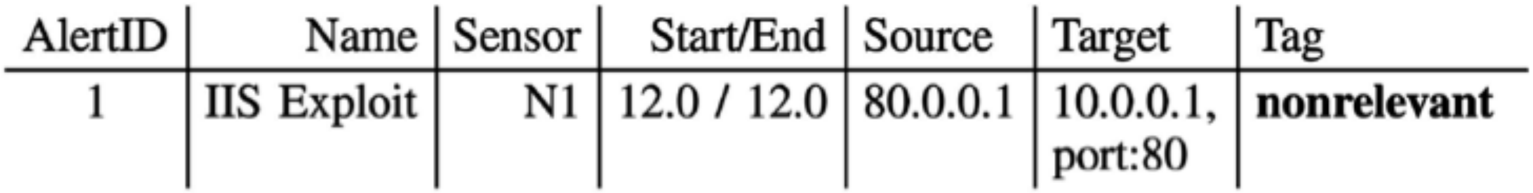


	MIT/LL 1999	MIT/LL 2000	CTV	Defcon 9	Rome AFRL	Honeypot	Treasure Hunt
Input Alerts	41,760	36,635	215,190	6,378,096	5,299,390	260,120	2,811,169
Output Alerts	39,094	36,631	215,113	4,565,029	5,299,390	260,120	2,808,595
Reduction	6.38%	0.01%	0.04%	28.43%	0.00%	0.00%	0.09%

Alert verification

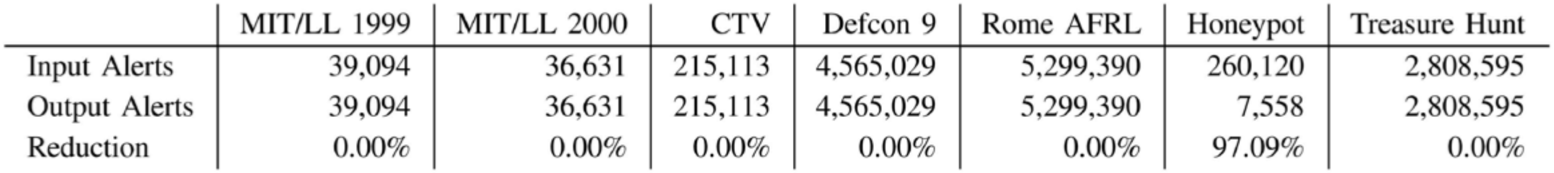
- True positive
- Irrelevant positive
- False positive
- Idea: extending intrusion detection signatures with an expected “outcome” of the attack
    - visible and verifiable traces left by attack
    - Example: temporary file, outgoing connection

Alert verification



AlertID	Name	Sensor	Start/End	Source	Target	Tag
1	IIS Exploit	N1	12.0 / 12.0	80.0.0.1	10.0.0.1, port:80	**nonrelevant**

Alert verification

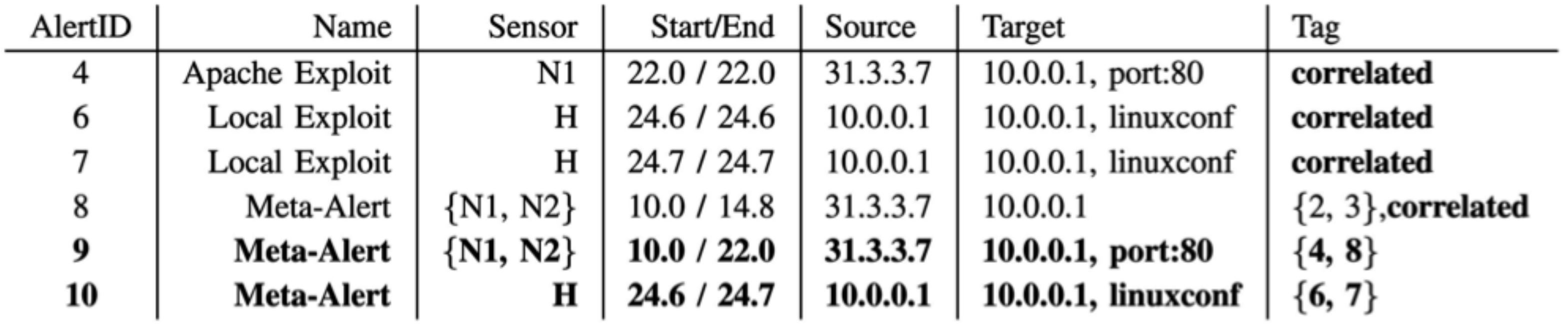


	MIT/LL 1999	MIT/LL 2000	CTV	Defcon 9	Rome AFRL	Honeypot	Treasure Hunt
Input Alerts	39,094	36,631	215,113	4,565,029	5,299,390	260,120	2,808,595
Output Alerts	39,094	36,631	215,113	4,565,029	5,299,390	7,558	2,808,595
Reduction	0.00%	0.00%	0.00%	0.00%	0.00%	97.09%	0.00%

Attack thread reconstruction

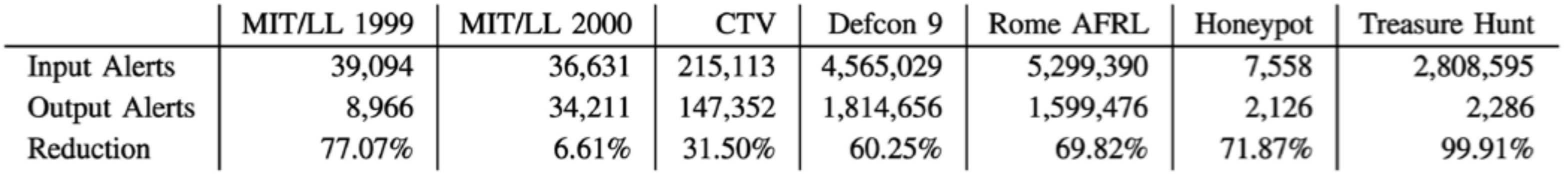
- Combines a series of alerts due to attacks by one attacker against a single target
- Idea: Merging alerts with equivalent source and target attributes in temporal proximity

Attack thread reconstruction



AlertID	Name	Sensor	Start/End	Source	Target	Tag
4	Apache Exploit	N1	22.0 / 22.0	31.3.3.7	10.0.0.1, port:80	**correlated**
6	Local Exploit	H	24.6 / 24.6	10.0.0.1	10.0.0.1, linuxconf	**correlated**
7	Local Exploit	H	24.7 / 24.7	10.0.0.1	10.0.0.1, linuxconf	**correlated**
8	Meta-Alert	{N1, N2}	10.0 / 14.8	31.3.3.7	10.0.0.1	{2, 3},**correlated**
9	**Meta-Alert**	**{N1, N2}**	**10.0 / 22.0**	**31.3.3.7**	**10.0.0.1, port:80**	**{4, 8}**
10	**Meta-Alert**	**H**	**24.6 / 24.7**	**10.0.0.1**	**10.0.0.1, linuxconf**	**{6, 7}**

Attack thread reconstruction

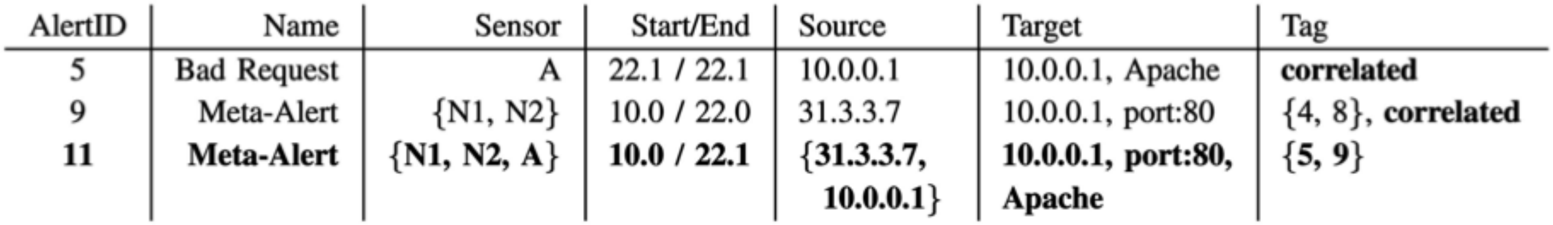


	MIT/LL 1999	MIT/LL 2000	CTV	Defcon 9	Rome AFRL	Honeypot	Treasure Hunt
Input Alerts	39,094	36,631	215,113	4,565,029	5,299,390	7,558	2,808,595
Output Alerts	8,966	34,211	147,352	1,814,656	1,599,476	2,126	2,286
Reduction	77.07%	6.61%	31.50%	60.25%	69.82%	71.87%	99.91%

Attack session reconstruction

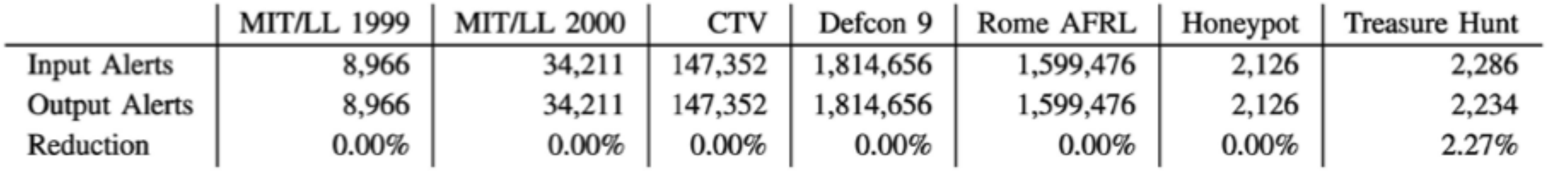
- Goal: Link network-based alerts to related host-based alerts
- Idea: Rough spatial and temporal correspondence between the alerts.

Attack session reconstruction



AlertID	Name	Sensor	Start/End	Source	Target	Tag
5	Bad Request	A	22.1 / 22.1	10.0.0.1	10.0.0.1, Apache	**correlated**
9	Meta-Alert	{N1, N2}	10.0 / 22.0	31.3.3.7	10.0.0.1, port:80	{4, 8}, **correlated**
11	**Meta-Alert**	**{N1, N2, A}**	**10.0 / 22.1**	**{31.3.3.7, 10.0.0.1}**	**10.0.0.1, port:80, Apache**	**{5, 9}**

Attack session reconstruction

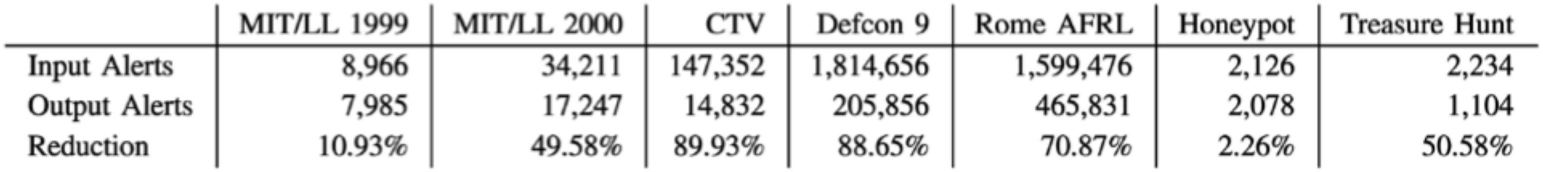


	MIT/LL 1999	MIT/LL 2000	CTV	Defcon 9	Rome AFRL	Honeypot	Treasure Hunt
Input Alerts	8,966	34,211	147,352	1,814,656	1,599,476	2,126	2,286
Output Alerts	8,966	34,211	147,352	1,814,656	1,599,476	2,126	2,234
Reduction	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.27%

Attack focus recognition

- Goal: identify hosts that are either the source or the target of a substantial number of attacks

Attack focus recognition

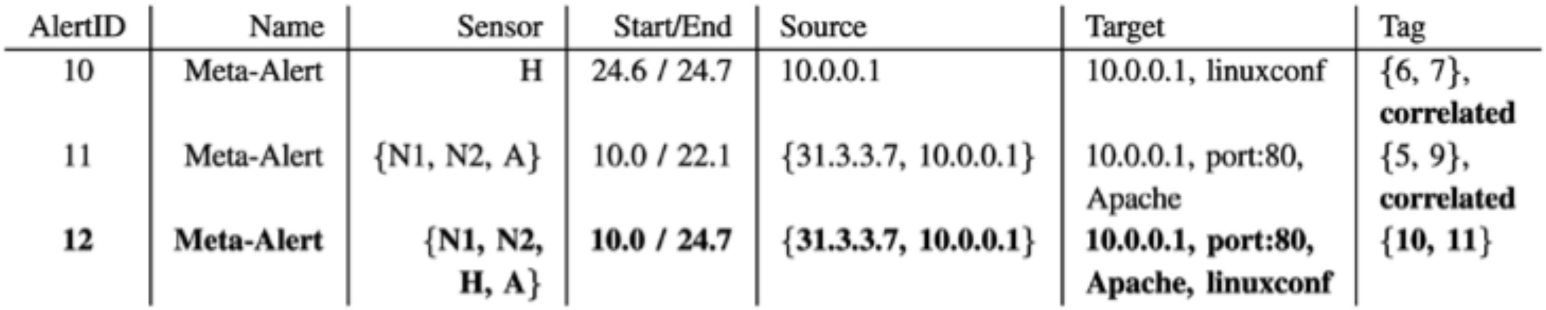


	MIT/LL 1999	MIT/LL 2000	CTV	Defcon 9	Rome AFRL	Honeypot	Treasure Hunt
Input Alerts	8,966	34,211	147,352	1,814,656	1,599,476	2,126	2,234
Output Alerts	7,985	17,247	14,832	205,856	465,831	2,078	1,104
Reduction	10.93%	49.58%	89.93%	88.65%	70.87%	2.26%	50.58%

Multistep correlation

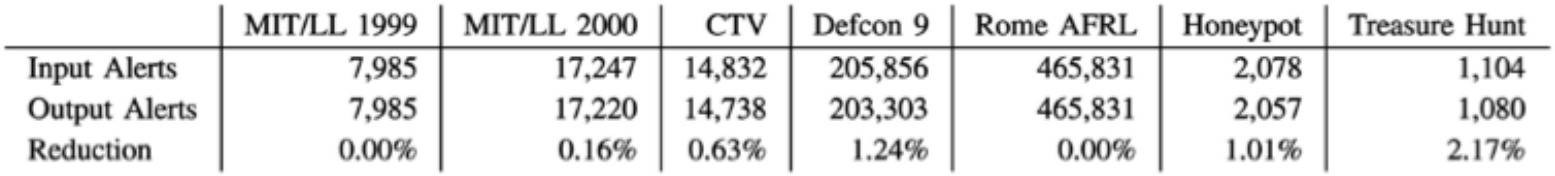
- Goal: identify high-level attack patterns that are composed of several individual attacks
- High-level attack signatures
  - Example: recon-breakin-escalate, island-hopping

Multistep correlation



AlertID	Name	Sensor	Start/End	Source	Target	Tag
10	Meta-Alert	H	24.6 / 24.7	10.0.0.1	10.0.0.1, linuxconf	{6, 7}, **correlated**
11	Meta-Alert	{N1, N2, A}	10.0 / 22.1	{31.3.3.7, 10.0.0.1}	10.0.0.1, port:80, Apache	{5, 9}, **correlated**
12	**Meta-Alert**	**{N1, N2, H, A}**	**10.0 / 24.7**	**{31.3.3.7, 10.0.0.1}**	**10.0.0.1, port:80, Apache, linuxconf**	**{10, 11}**

Multistep correlation



	MIT/LL 1999	MIT/LL 2000	CTV	Defcon 9	Rome AFRL	Honeypot	Treasure Hunt
Input Alerts	7,985	17,247	14,832	205,856	465,831	2,078	1,104
Output Alerts	7,985	17,220	14,738	203,303	465,831	2,057	1,080
Reduction	0.00%	0.16%	0.63%	1.24%	0.00%	1.01%	2.17%

Course summary

- Introduction, infosec goals, failure of prevention & reactive defense
- Botnet topologies, botnet detection
- Host-based insider threat detection
- Biometrics
- Web security
- Adversarial machine learning
- Deep packet inspection
- Cautionary notes
- Multi-classifier systems: supervised and one-class
- Polymorphism
- Phishing detection
- Alert correlation
- Industry perspectives
- Student presentations

Final thoughts

- Trends in security
- Thanks and Good luck!

Reference

- “A Comprehensive Approach to Intrusion Detection Alert Correlation”, Valeur et al, 2004