

Network Intrusion Detection Systems

False Positive Reduction
Through Anomaly Detection

Joint research by
Emmanuele Zambon & Damiano Bolzoni

- Introduction: the NIDS problems
- A strategy for reducing false positives rate
- Outbound traffic validation issues
- POSEIDON: a payload-oriented anomaly detection system
- Correlation issues: input-side alerts and output traffic analysis
- APHRODITE: the architecture for FP reduction
- Experiments
- Conclusions & future work
- Questions
- References

Agenda

Network Intrusion Detection Systems, no matter if they are Signature or Anomaly based, have in common some problems

False
Positives

NIDS problems connected with false alerts

The **number of alerts** collected by an IDS can be very large (15,000 per day per sensor).

The **number of FP** alerts is very high (thousands per day).

Reducing the FP rate often, causes **worse NIDS reliability**.

The task of filtering and analyzing alerts **must be done manually**.

All of these problems cause the final user of NIDS, the security manager, to:

- an **overload** of work to recognize true attacks from NIDS mistakes
- **loose confidence** in alerts
- **lower the defences** level to reduce FP number

NIDS problems

Tuning the NIDS can solve some of the FP problems, but...

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS  
$HTTP_PORTS (msg: "WEB-MISC http directory  
traversal"; flow:to_server,established;  
content:"../"; reference:arachnids,297;  
classtype:attempted-recon; sid:1113; rev:5;)
```

False Positive

```

```

TUNING IS NOT ENOUGH!

We believe that the problem with current NIDS is that they ignore roughly half of the network traffic

FP's occur when the NIDS fails to consider the legitimate sampled traffic as an attack.

We need a way to confirm that an attack is taking place, **before raising any alert.**

Some considerations ...

When an attack takes place, it is likely to produce some kind of **unusual effect** on the target system.

On the other hand, if the data flow is licit, there will be no unusual effect on the target system.

Considering a network environment we can observe the reaction of monitored systems by examining the **outgoing data** flowing from those systems in response of an extern solicitation.

Current NIDS **only** consider **incoming** requests of monitored systems, as attacks always come from those ones.

To increase accuracy on NIDS ability to distinguish real attacks, we need to introduce **correlation** between incoming and outgoing data.

A strategy for reducing FP rate

In general, all real attacks modify the information flow between the monitored system and the systems with which it is dialoguing with.

Classes of attacks

Consequences

Attacks of interruption

Attack on the availability of the system

When an attack causes the interruption of one or more services in a system, or even causes a system failure, all communications are stopped. Observing output network traffic we will see no more data flowing outside the monitored system.

Attacks of interception

Unauthorized access to a system

Unauthorized access to a system is mostly done to gain information they wouldn't normally get by the system. If an attempt of attack is done, and the system reacts denying the information disclosure, it will usually send some kind of error message, or no data at all.

Attacks of modification

Attack on the integrity of the system

When an attack causes the modification of the information provided by a system, the behaviour of the system itself will be altered, causing it to alter its normal information flow.

Attacks of fabrication

Degrades the authenticity of the system

If an unauthorized party gains access to the system and inserts false objects into it, it degrades the authenticity of the system. This causes a deviation in the normal behavior of the system, reflecting in the alteration of the usual output of the system itself.

Attacks modify normal information flow

Validation of output traffic for a system is more complex than input validation.

Problems in output traffic validation

Every instance of an application in a system has a **different kind** of output traffic, according to the information it contains.

There is a number of ways a system can react to an attack. Even if the same attack is carried out on two different system, the **reaction won't be the same**.

How can we **associate input traffic with output**?
How much must we wait to see the response to a suspicious request?

A signature-based tool is not suitable for output validation.
We need **anomaly detection!**

We need a **correlation engine** to associate correctly input suspicious request with appropriate responses.

Problems in output traffic validation

Anomaly detection is more suitable to adapt to the specificity of output traffic inspection.

Advantages

It can be **trained** to recognize the “normal” output traffic of a system / protocol.

The “normal” output traffic model, built during training phase, **fits at best** the specific features of the system it has been trained on.

Output traffic that is found to be anomalous can be used to **confirm** the raising of an alert.
Anomalous traffic is that one that deviates from his normal behaviour.

An output validator built with anomaly detection techniques can confirm alerts raised even if the attack is **0-day**.

Disadvantages

The training phase takes generally a **lot of time** and is crucial for the performances during the detection phase.

If some features of the system are changed, the “normal” traffic can change and **models need to be updated**.

To achieve output traffic validation, according with the previous considerations, we designed POSEIDON, a NIDS based on the anomaly-detection approach

POSEIDON stands for:
Payl Over Som for Intrusion DetectiON

Starting from the good results achieved by K. Wang and S. Stolfo with their IDS (**PAYL**) we built a two-tier NIDS that improves the number of detected attacks using a Self Organizing Map (**SOM**) to pre-process the traffic.



Main Features

Network-oriented.

Payload-based. It considers mainly the payload of the traffic it inspects.

Two-tier architecture.

Developed and tested for **TCP traffic.**

PAYL is the base from which we started building our anomaly detection engine

PAYL features

Anomaly-based engine that uses **full payload** data to detect anomalies.

To characterize traffic profiles only a **few other features** is used:

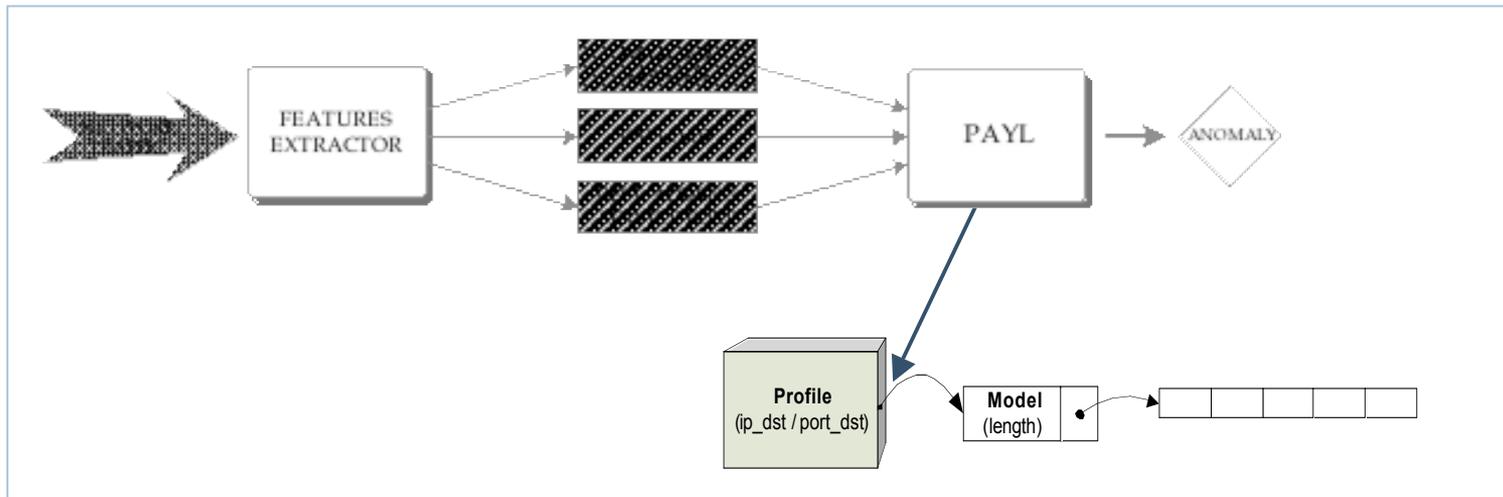
- monitored host **IP** address
- monitored **Service Port**
- payload **length**

Enhanced by post model-building **clustering**

Benchmarked with **reference dataset** (DARPA 1999)

For anomaly detection a slightly modified **Mahalanobis distance function** is used.

High detection rate. Low false positives rate.



PAYL (Wang and Stolfo, 2004)

PAYL classification method presents some weaknesses that compromise the quality of normal traffic models

PAYL classification weaknesses

Data with different contents can be **clustered in the same class**.

Similar data can be clustered in **two different classes** because the length presents a **small difference**.



PAYL classification does not evaluate properly INTER-CLASS SIMILARITY.

Is it possible to enhance PAYL classification model?

We need **unsupervised** classification

We must classify **high-dimensional data** (the full payload data)

Enhancing PAYL

T. Kohonen, in 1995, describe a data visualization technique which reduce the dimensions of data through the use of self-organizing neural networks

KEY features

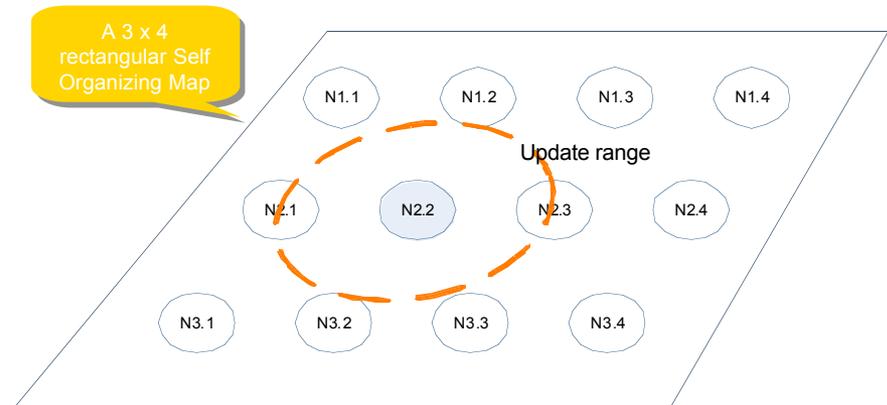
Competitive networks with **unsupervised** learning.

SOM training phases:

- Initialization
- Get Best Matching Unit (**BMU**)
- **Update** scaling neighbours

New samples are used to update network with **reducing neighbourhood influence** over time

It is possible to determinate the quality of trained network by **quantization error**.



Advantages

Unsupervised and suitable for **high dimensional** data

Suitable to **detect anomalies** in network traffic (*quantization error*)

Benchmarked against other clustering algorithms (K-means, K-medoids)

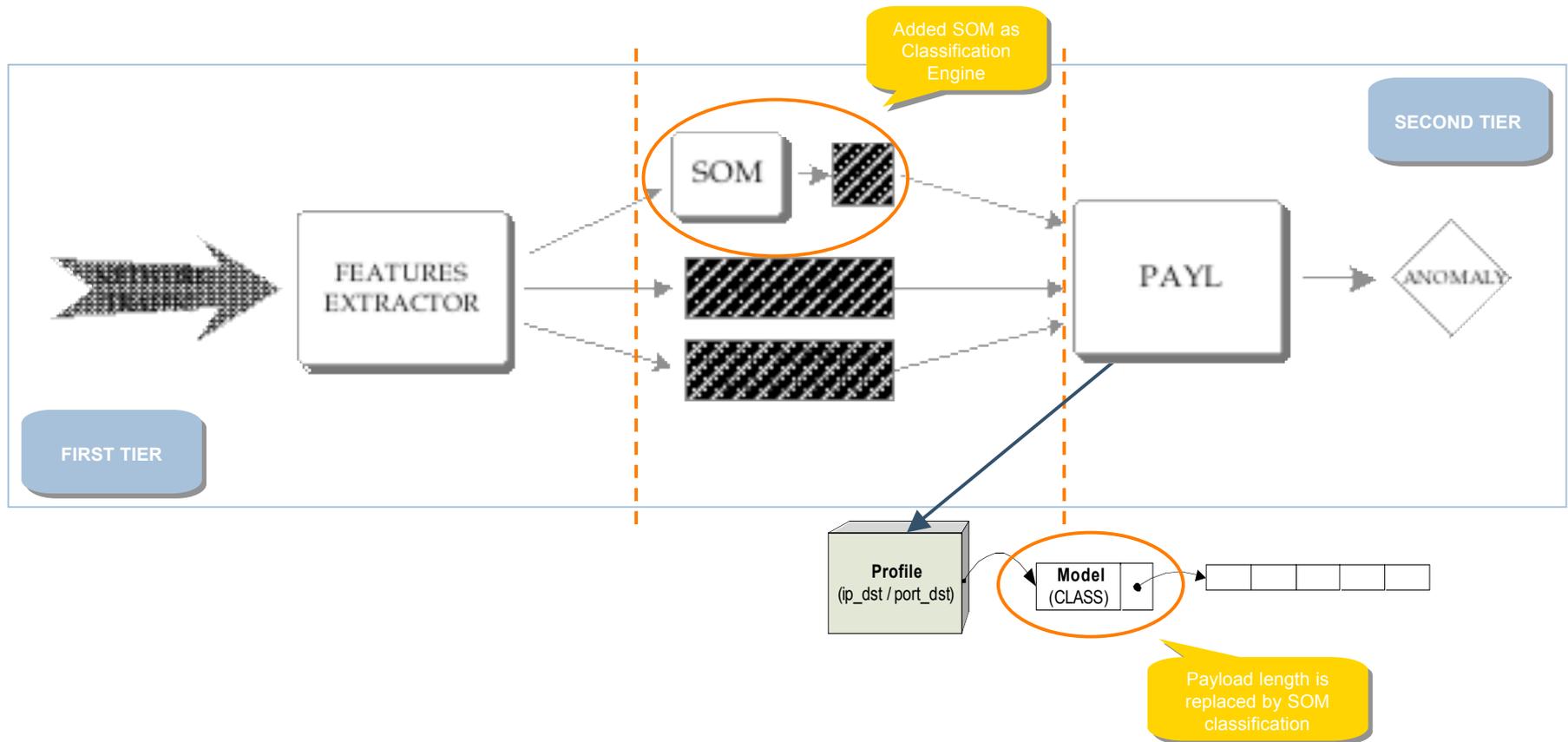
Disadvantages

Requests a **training** phase

Too many false positives (SOM **does not evaluate** properly **intra-class similarity**)

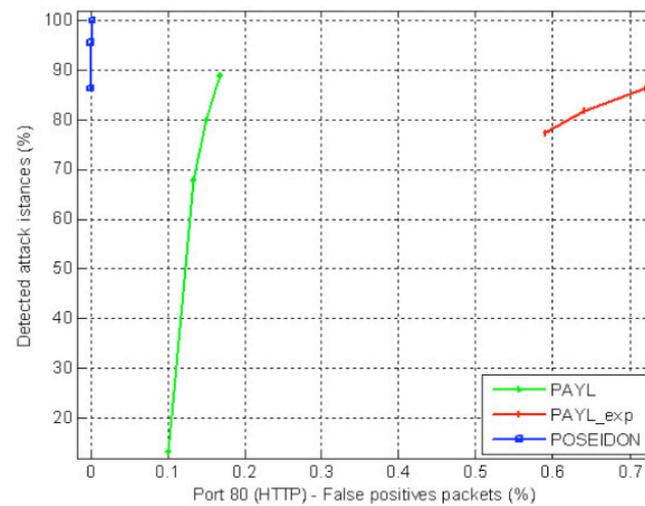
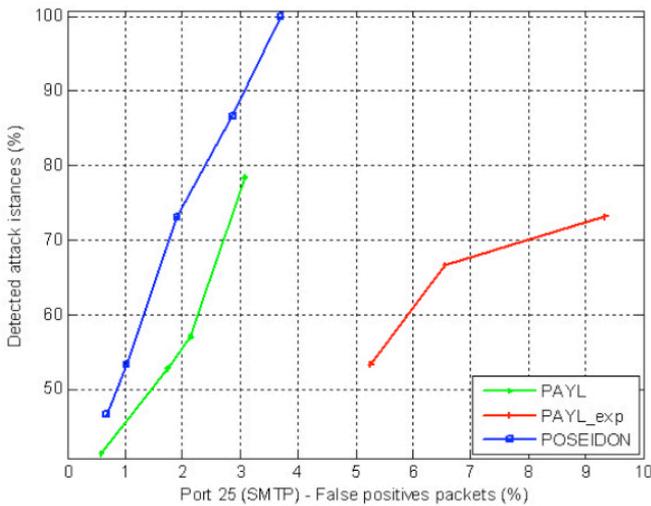
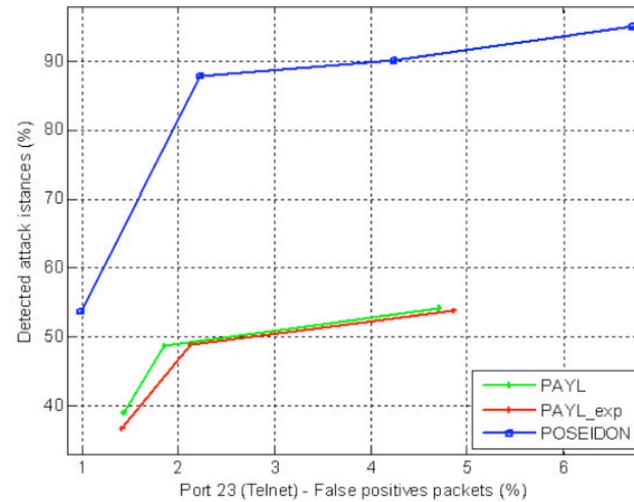
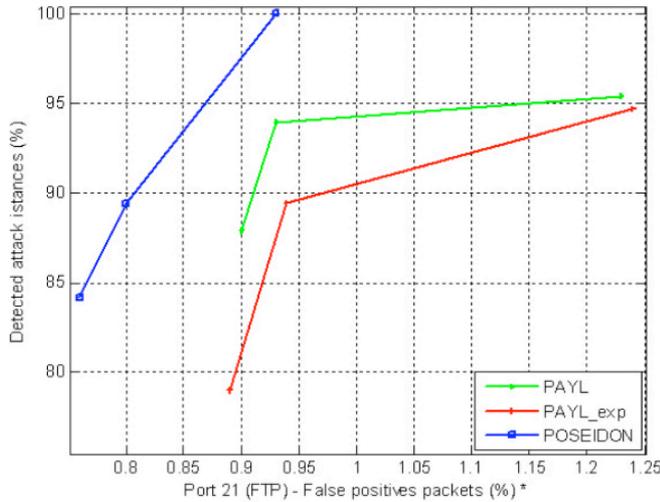
SOM – Self Organizing Maps

Using SOM to classify payload, according to service port and monitored IP address, improves PAYL model building phase



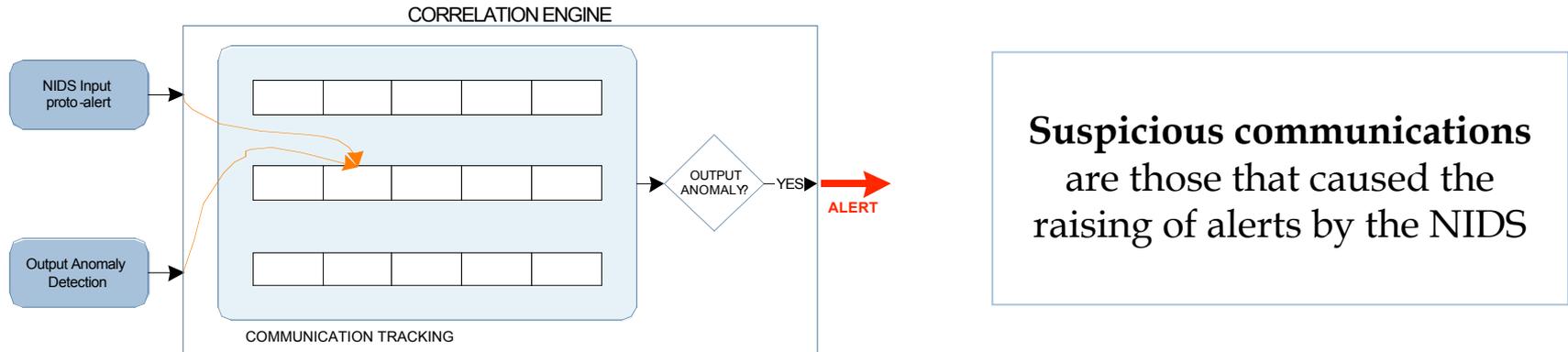
POSEIDON - Architecture

POSEIDON overcomes PAYL on every benchmarked protocol



POSEIDON – Test Results

When a proto-alert is raised, the correlation engine considers the output validator results and forwards the alerts only if there is an output anomaly... with some exceptions.

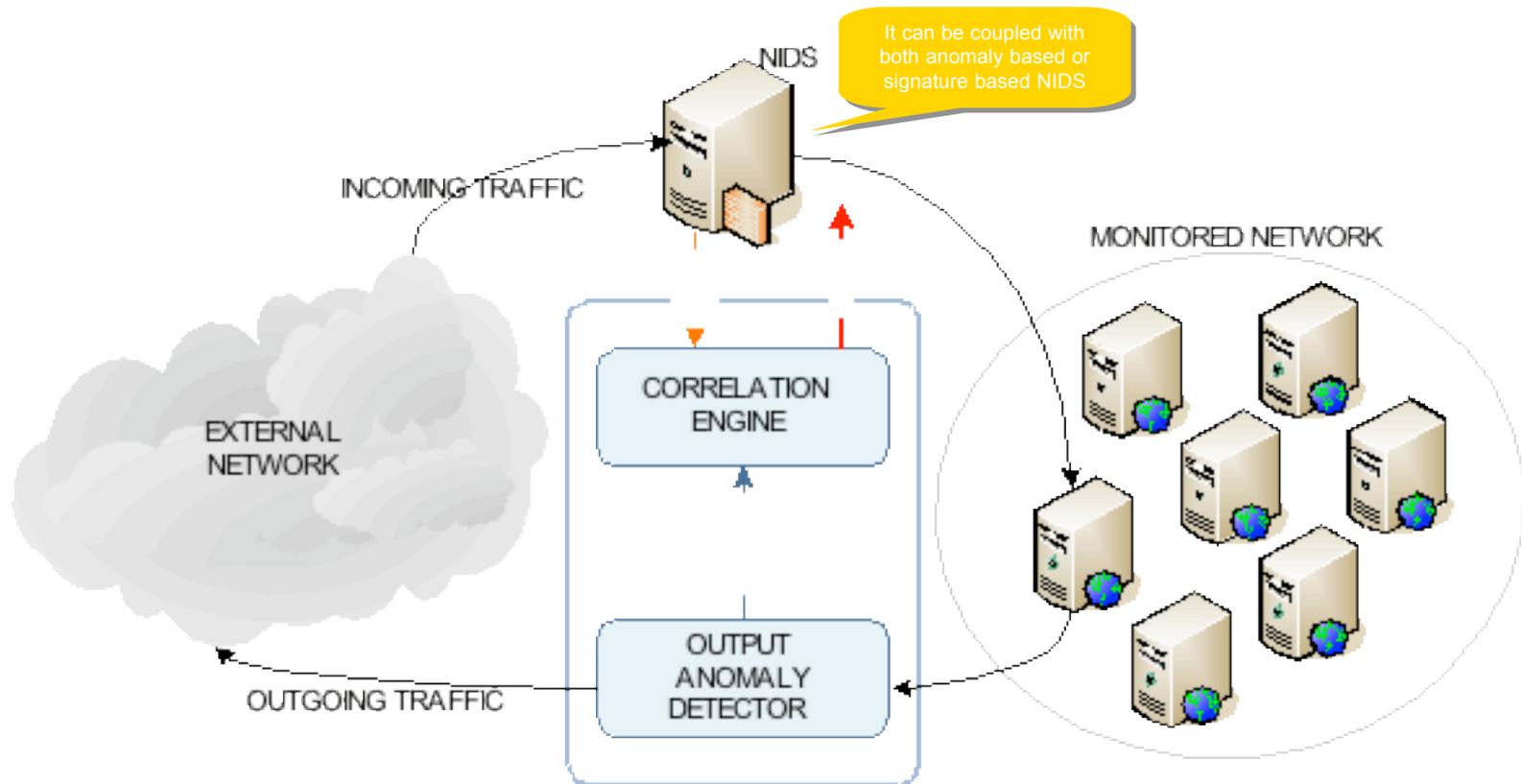


Exception	Description
Missing output response	There could be an interruption attack (DoS). The alert is considered as a True Positive and forwarded.
Alarm magnitude	If the NIDS is anomaly based it can indicate the magnitude of the alert. If the alert magnitude is high, the alert can be considered as TP even if no suspicious output is found.
Number of alarm-raising packets	Number of alerts directed to a single endpoint are counted for a given time frame. If the count is very high, new alerts will be considered TP even if no suspicious output is found.

Need to set an appropriate application timeout

Correlation issues

APHRODITE is the architecture that combines the correlation engine and the output anomaly detector.



APHRODITE – High Level Architecture

We benchmarked APHRODITE using two different datasets, with both signature-based and anomaly-based NIDS

NIDS

We coupled APHRODITE with the well known open NIDS **Snort**:

- *signature-based*
- *totally open* (even the signature database)
- detection rule set is *configurable*

We also used **POSEIDON** as inbound traffic IDS:

- *anomaly-based*
- implementation *available*

Datasets

The first dataset we used was **DARPA 1999**:

- it has been designed and is widely used for *IDS benchmarking*
- allows one to *duplicate and validate* experiments
- attacks are *labelled*
- has been criticized because of the *unrealistic nature* of some data parameters

To make more exhaustive the tests, we used a second, **private dataset**:

- contains *5 days of HTTP traffic* collected from a public network
- no attack was *injected*
- attack were found and validated by *manual inspection* and NIDS processing

APHRODITE - Test Methodology

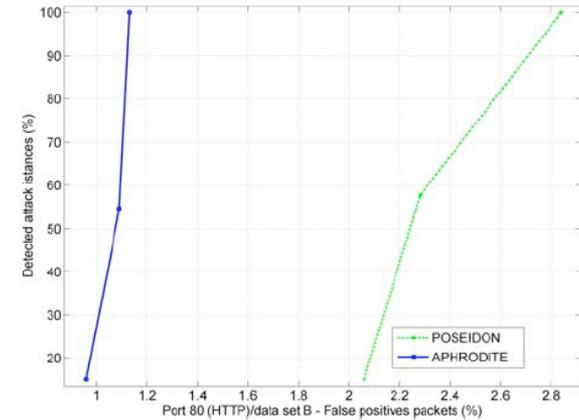
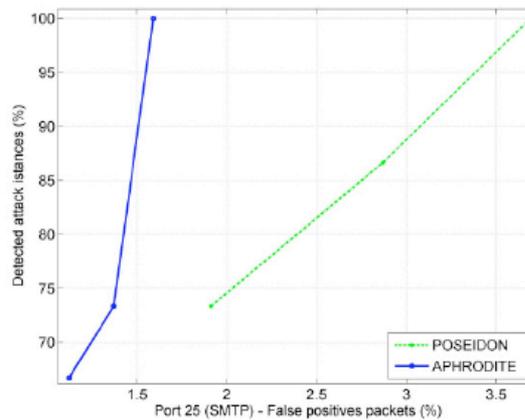
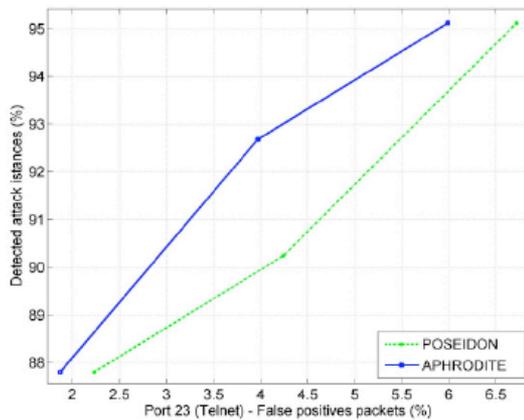
APHRODITE achieves a substantial improvement on the stand-alone systems

DARPA 1999

PROTOCOL		SNORT	SNORT + APHRODITE	POSEIDON	POSEIDON + APHRODITE
HTTP	Alerts	4318	3363	873235	873220
	DR	59,9 %	59,9 %	100 %	100 %
	FP	599 (0,068%)	5 (0,00057%)	15 (0,0018%)	0 (0,0%)
FTP	Alerts	904	336	3529	694
	DR	31,75 %	31,75 %	100 %	100 %
	FP	875 (3,17%)	317 (1,14%)	3303 (11,31%)	373 (1,35%)
Telnet	Alerts	1275	889	65832	54093
	DR	26,83 %	26,83 %	95,12 %	95,12 %
	FP	391 (0,041%)	6 (0,00063%)	63776 (6,72%)	56885 (5,99%)
SMTP	Alerts	2	-	10239	6072
	DR	13,3 %	-	100 %	100 %
	FP	0 (0,0%)	-	6476 (3,69%)	2797 (1,59%)

Private dataset

PROTOCOL		POSEIDON	POSEIDON + APHRODITE
HTTP	Alerts	1739	830
	DR	100 %	100 %
	FP	1683 (2,83%)	774 (1,30%)



APHRODITE – Test results

Conclusions:

- The experiments show that our modification to PAYL **improves the detection rate** and **reduce** sensibly **false positives rate**.
- We strongly believe that this result has been achieved by **replacing** the original **PAYL classification method** with a new algorithm (based on self-organizing maps).
- APHRODITE determinates a substantial **reduction of false positives**.
- Reduction of false positives **does not introduce extra false negative**.
- APHRODITE is still effective also when it is **not trained optimally** (in case of quick setup without an accurate tuning phase during training).

Future work:

- Make OAD updateable *without a new complete training phase*.
- Make the system able to *adapt itself to environment changes* in an automatic way.
- Automate the phase of *threshold computation*.

ANY QUESTION

?

Questions & Answers

- T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, 1995. (Second Extended Edition 1997).
- K. Wang and S. J. Stolfo. Anomalous Payload-Based Network Intrusion Detection. In E. Jonsson, A. Valdes, and M. Almgren, editors, *RAID '04: Proc. 7th symposium on Recent Advances in Intrusion Detection*, volume 3224 of *LNCS*, pages 203–222. Springer-Verlag, 2004.
- W. Stallings. *Network Security Essentials: Applications and Standards*. Prentice-Hall, 2000.
- Manganaris, S., Christensen, M., Zerkle, D., Hermiz, K.: A Data Mining Analysis of RTID alarms. *Computer Networks: The International Journal of Computer and Telecommunications Networking* **34**(4) (2000)
- Axelsson, S.: The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **3**(3) (2000)
- Snort NIDS. Official web site URL: <http://www.snort.org>.

References