# Reduce false positive using expert system on IDS

***Raghad Mohammed Hadi**

***Management and Economey collage / University of Al mstansuraia**

## Abstract

When an attacker tries to penetrate the network, there are many defensive systems, including intrusion detection systems (IDSs).Intrusion Detection refers to the process of monitoring the system for unauthorized access incidents which can be the violation of the security policy, system use policy, or any other security standards, An Intrusion Detection System (IDS) is software that implements the intrusion detection process. But it has weakness that cannot provide a clear idea to the analyst because of the huge number of false alerts generated by these systems. This weakness in the IDS made us think about providing improvements to it for reduces false alerts, detects high level patterns of attacks, and increases the meaning of occurred incidents. So that further research in this area will be motivated objectively to fulfill the gaps exists till now. In this paper we present expert system approaches to reduce the number of false positives alert in intrusion detection using alert processing.

**Key-Words**: -Network security, intrusion detection, alert correlation, False alarm, attacks**.**

# تقليل الايجابية الكاذبة باستخدام نظام خبير لنظام الكشف عن المتطفلين

*** رغد محمد هادي**
*** كلية الادارة والاقتصاد / الجامعة المستنصرية**

**الخلاصة**

عند محاولة أحد المهاجمين اختراق الشبكة، فان هناك العديد من الأنظمة الدفاعية، بما في ذلك أنظمة الكشف عن التسلل (Intrusion detection system) . يشير الكشف عن التسلل الى انه عملية رصد النظام لحوادث الوصول غير المصرح به والتي يمكن أن تكون مخالفة لسياسة الأمن، وسياسات استخدام نظام ، أو أي معايير الأمنية الأخرى، نظام كشف عن التسلل (الهوية) هو البرنامج الذي ينفذ عملية الكشف عن التسلل. لكنه يملك الضعف الذي يتمثل بعدم توفير فكرة واضحة للمحلل نظرا لوجود عدد كبير من الانذارات الكاذبة التي تولدها هذه النظم. هذا الضعف في IDS جعلنا نفكر في تقديم تحسينات له ليقلل من الانذارات الكاذبة، والكشف عن أنماط مستوى عال من الهجمات، ويزيد من معرفة الحوادث التي وقعت. بحيث اصبح دافعا لإجراء مزيد من البحوث في هذا المجال لتحقق من الثغرات الموجود فيه. في هذه االبحث تم استخدام نظام خبير للحد من عدد من ايجابيات كاذبة (false alert) في حالة تأهب كشف التسلل باستخدام معالجة alerts..

**الكلمات المفتاحية:**

**امنية الشبكات ، الكشف عن المتسللين ، ربط التنبيهات ، انذار خاطى ، الهجوم.**

## 1. Introduction

Intrusion detection (ID) is a blanket term for detecting inappropriate, harmful, or anomalous activity on a computer or network. in the several reports, there are many sources of intrusion information that can be used to illuminate what actually happened during a system compromise. However, there are also "real time" intrusion detection systems that use current process, network, and state information to determine if unusual activity is currently in progress. There are many different sorts of intrusion detection methods, ranging from the historical to the modern. In this paper, we will take a look at the two methods and philosophies of detecting intrusive or unusual events. This includes looking at both the theory behind intrusion detection, as well as considering current products on the market.[1]

## 2. Intrusion Detection system (IDS):

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to bypass the security mechanisms of a computer or network ("compromise the confidentiality, integrity, availability of information resources") **Intrusion Detection** System (IDS): combination of software and hardware that attempts to perform intrusion detection raise the alarm when possible intrusion happens.[2]

## 3. Expert Systems

Expert systems are programs that try to emulate human expertise and problem solving abilities through use of a technique called "rule-based"

programming. Rule-based programming makes use of heuristics, or "rules of thumb", to specify actions to perform when specific patterns of data are encountered.[3] we use one technique of Rule-based programming called **particle swarm optimization (PSO).**

PSO was originally developed by a social-psychologist J. Kennedy and an electrical engineer R. Eberhart in 1995 and emerged from earlier experiments with algorithms that modeled the "flocking behavior" seen in many species of birds. Where birds are attracted to a roosting area in simulations they would begin by flying around with no particular destination and in spontaneously formed flocks until one of the birds flew over the roosting area [10]. PSO has been an increasingly hot topic in the area of computational intelligence. It is yet another optimization algorithm that falls under the soft computing umbrella that covers genetic and evolutionary computing algorithms as well [11].

While Rules are composed of an if portion and a *then* portion. The if portion of a rule is a series of patterns which specify the data that causes the rule to "activate". The *then* portion is a set of actions to be executed upon this activation.[3]

The above rule, when provided with a fact describing the type of intrusion detection system, and a fact describing an attack itself, will *assert* or add a new fact to the system, containing information about the intrusion detection system, the type of attack, alert CID, and destination IP address. The *if* portion of the rule is specified by what appears before the arrow (=>), and the *then* portion is what appears after wards**[ 4 ].**

## 4. Network Threats

Attacks (intrusions) on the network are actions that attempt to bypass security mechanisms of computer systems. They are caused by:

1) Attackers accessing the system from Internet.
2) Insider attackers - authorized users attempting to gain and misuse non-authorized privileges.
3) Security mechanisms always have inevitable vulnerabilities.
4) Current firewalls are not sufficient to ensure security in computer networks.
5) "Security holes" caused by allowances made to users/programmers/administrators.
6) Multiple levels of data confidentiality in commercial and government organizations needs multi-layer protection in firewalls.[5]

## 5. Implementation toolset

The tools use to implementation proposed system in world web site must use different types of tools, one of the tools is language, web programming language used to compose a system, the language types are:

•**ASP.NET(Active Server Pages)** are HTML page, which include scripting and create active web server applications,ASP.NET language is very important to build web application and different from the previews language to have many tools to build site and can used the visual basic or JavaScript .NET language .

- **C++** is an extendable language in which we can define new data types and "objects" in such a way that they act like part of the standard language.
- **Sno**rt rule #885 looks for the presence of the string "/bash" in packet content. This rule may alerts the soc about a hacker that is trying to get a bash shell on a target machine.

## 6. The Proposed System Platform Requirements:

The platform is foundation of the software solution and should be presented first. The core component will be indicted, piecing it all together in overall architecture, with some though about communication, **figure (1)** shows the proposed system's platform, the proposed system content the following parts:

### 6.1 IDS (Intrusion detection system):

Intrusion is the sequence of the set of related activity which perform unauthorized access to the useful information and unauthorized file modification which causes harmful activity as we say above. Intrusion detection system deal with supervising the incidents happening in computer system or network environments and examining them for signs of possible events, which are infringement or imminent threats to computer security, or standard security practices.[6]

Snort is an open-source, lightweight, network intrusion detection system. There are three main ways in which Snort can be used: as a packet sniffer, a packet logger, and a Network Intrusion Detection System (NIDS) in the proposed system we use Snort as NIDS.[4] see figure (4).

### 6. 2 Alerts:

Alert is defined as an alarm generated by Intrusion Detection system (IDS), to notify interested parties of interesting event. An event is a low level entity analyzed by IDS. Single event can cause multiple alerts and it can be represented in mathematical expression as below:

Event = {alert1, alert2, alert3…..alertn} [7]

There are many terms that are usually misunderstood and should be differentiated between them that are *event, alert* and *alarm*. An *event* is a low level entity that is analyzed by the IDS, whereas an *alert* is generated by the IDS to notify parties of interesting events. A single event can cause many alerts (that is a problem) especially in a NIDS environment, and a single alert can describe a set or sequence of events [8]. Every alert is suspicious but an event is not necessarily suspicious. An *alarm* is the user interface mechanism by which a user manages an alert [9]. **See figure (3)**

## 6.3 Alerts processing:

In this part we will discuss an alerts-processing system, a system which uses clusters generated, labels them and applies them to future alerts. As discussed, there are two possible modes of "application" of this alert-processing system: (i) cluster processing mode and (ii) filtering mode. Cluster processing Mode (CM) in the first mode, the alert cluster membership is evaluated and the matching cluster is subsequently used to construct additional features. Those features can be used by the ID analyst to facilitate alert classification. This step is illustrated in alert clustering algorithm section.

## 6.4 Alerts correlation:

Intrusion alert correlation is multi-step processes that receives alerts from heterogeneous log resources as input and produce a high-level description of the malicious activity on the network.**[7]**

In the proposed system we may say that correlation is finding relationships between alerts generated by a single (or multiple) data sources and coupling this information with additional knowledge, i.e., a way which can be easily understood and processed by a human analyst and helping him discover attacks and incidents.

Correlation in proposed system uses rules, manually programmed or derived automatically from configuration parameters. We uses expert system program to describe an algorithm and architecture of an Aggregation and Correlation (AC) Component. The goal of the algorithm is to form groups of alerts by creating a small number of relationships. We define two kinds of relationship: an alert correlation relationship and an aggregation relationship.

In the alert correlation step we proposed algorithm consists of two sequential steps: In the first step, alerts are verified (that they do not contain invalid information e.g., invalid timestamps) and unified (e.g., different representations of hostnames and IP addresses or port numbers and services).

The second step deals with alerts which are linked with each other. It uses system expert system PSO as we say above to group duplicates (i.e., semantically equivalent alerts) and consequences (alerts appearing in a given order) and this step is illustrated in expert system algorithm section.

## 7. Alert clustering algorithm:

**Input:** set of alerts & cluster array Pi.

**Output:** send defined alert with its features to security analyst.

**Begin**

**Step1**: collected alerts in the most recent week, as a set of clusters {P1, P2, . . . , Pn}.

**Step2**: match Alerts in the clusters with a set of incidents {I}, based on which cluster specific aggregates and features are calculated.

**Step3**: **while** (alert array not empty) and (predefined time has elapsed (e.g. week) **do**

**begin**

each new alert Ai is matched with a set of clusters {P1, P2, . . . , Pn}.

**check if** new alert Ai is equal to one of a cluster Pi, **then**

the cluster-specific features are associated with new alerts.

End while

**Step4**: send new alert Ai with its features to security analyst.

**End.**

## 8. Expert system algorithm:

Expert system program in proposed system attempt to Inference by legitimate user because an intrusion user attempting to obtain unauthorized data from a database through aggregation and inference might retrieve more records than usual if this user is authorized, then we used PSO algorithm as one of these stopping criterions to calculate the fitness value (unauthorized data). And there are observed values of (ti) and desired output values of (fi). These two values have to be compared, if they are closed to each other then the fitness is good, else the algorithm must continue its calculations until this condition is satisfied or its false alert.

The corrections to the weights are selected to minimize the residual error between ti and fi output. The Mean Squared Error (MSE) is one solution for the comparison process:

MSE = [1/n ∑ (ti - fi)(ti – fi)] for i=1 to n … (1)

Where n is the number of the compared categories.

### 8.1 PSO Algorithm

The PSO algorithm depends in its implementation in the following two relations:

vid=w*vid+c1*r1*(pid-xid)+c2*r2*(pgd-id) ...(2a)

xid = xid + vid …(2b)

where c1 and c2 are positive constants, r1 and r2 are random function in the range [0,1], xi=(xi1,xi2,…,xid) represents the ith particle; pi=(pi1,pi2,…,pid) represents the best previous position (the position giving the best fitness value(unauthorized data or Trojan horse or may be viruses) of the ith particle; the symbol g represents the index of the best particle among all the particles in the population, v=(vi1,vi2,…,vid) represents the rate of the position change (velocity) for particle i [2].

The original procedure for implementing PSO is as follows:

**Input**: set a population of particles with random positions and velocities on d-dimensions in the problem space.

**Output**: calculate the fitness value (unauthorized data)

**Begin**

**Step1**: For each particle, evaluate the desired optimization fitness Function in d variables.

**Step2**: **while** a criterion is not met **do**

**Begin**

1) Compare particle's fitness evaluation with its pbest.
2) If current value is better than pbest, then set pbest equal to the current value, and pi equals to the current location xi.
3) Identify the particle in the neighborhood with the best success so far, and assign it index to the variable g.
4) Change the velocity and position of the particle according to equation (2a) and (2b).

**End**

**End.**

Like the other evolutionary algorithms, a PSO algorithm is a population based on search algorithm with random initialization, and there is an

interaction among population members. Unlike the other evolutionary algorithms, in PSO, each particle flies through the solution space, and has the ability to remember its previous best position, survives from generation to another. The flow chart of PSO algorithm is shown in figure (2) .

## 9. Conclusions:

**1.** Building Intrusion Detection Systems is still difficult and expensive because of a lack of structured methodology. Lack of agreement on intrusion detection techniques and tools hinders the development of such a methodology.

**2.** Many Intrusion Detection methods are computationally expensive,
and need extensive system profiles and libraries of attack signatures. Also, somekinds of Intrusion Detection Systems are also implemented using expert systems,causing a high runtime overhead and limiting the representation of possiblerelationships between events.

**3.** Most intrusion detection systems have been written for single environments, and have proved difficult to port to others. This limits the reuse and retargeting of intrusion detection systems. It is difficult to get rid of these specific system customizations, as some are currently necessary to detect certain kinds of attacks in certain computing environments.

**4.** It is often difficult to integrate newer and better intrusion detection techniques with older existing intrusion detection systems.

**5.** Maintenance of intrusion detection systems requires a high level of security knowledge from the administrator. Some specialized non-securityknowledge necessary may include expert system rule language, and statisticalcalculating methods. This specialized knowledge makes administering intrusiondetection systems difficult and costly for the average system administrator.

## 10. References:

**1.** Advisors : dr. M.E.M. Spruit (TU Delft), ir. R. Prins (Fox-IT), " The Meta-Alert Correlation Engine",, Faculty of Information technology and Systems Technical Informatics Delft July 11, 2003.

**2.** Paul Dokas, LeventErtoz, Vipin Kumar, AleksandarLazarevic, JaideepSrivastava," Data Mining for Network Intrusion Detection ", Pang-NingTan,2004.

**3.** Gary Riley, "What is CLIPS?", http://www.ghg.net/clips/WhatIsCLIPS.html

**4.** "OpenBSD FAQ: Introduction to OpenBSD",http://www.openbsd.org/faq/faq1.html.

**5.** AleksandarLazarević, JaideepSrivastava, Vipin Kumar " Data mining for intrusion detection " , Tutorial on the Pacific-Asia Conference on Knowledge Discovery in Databases 2003.

**6.** Dorothy Denning, "An Intrusion Detection Model", IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, February 1987, 222-232.

**7.** RobiahYusof, SitiRahayuSelamat, Shahrin Sahib, " Intrusion Alert Correlation Technique Analysis for Heterogeneous Log, 2008.

**8.** Qin X. and Lee W., "Statistical causality of INFOSEC alert data," Proceedings: Recent Advances in Intrusion Detection, LNCS 2820; Springer-Verlag, pp. 73-93, 2003.

**9.** Valeur F., Vigna G., Kruegel C. and Kemmerer R. A., "Comprehensive approach to intrusion detection alert correlation," IEEE Transactions on Dependable and Secure Computing 1 (3), pp. 146-169, 2004.

**10.** Amin, S.M. and Rodin E. Y., "Neurocontrol of Nonlinear System via Local Memory Neurons", Math. & computer Modeling vol. 27, No. 3, pp. 65-92, 1998.

**11.** Pomeroy P. "An Introduction to Particle Swarm Optimization", Article, Mar, www.adaptiveview.com, pages. 1-7, 2003.

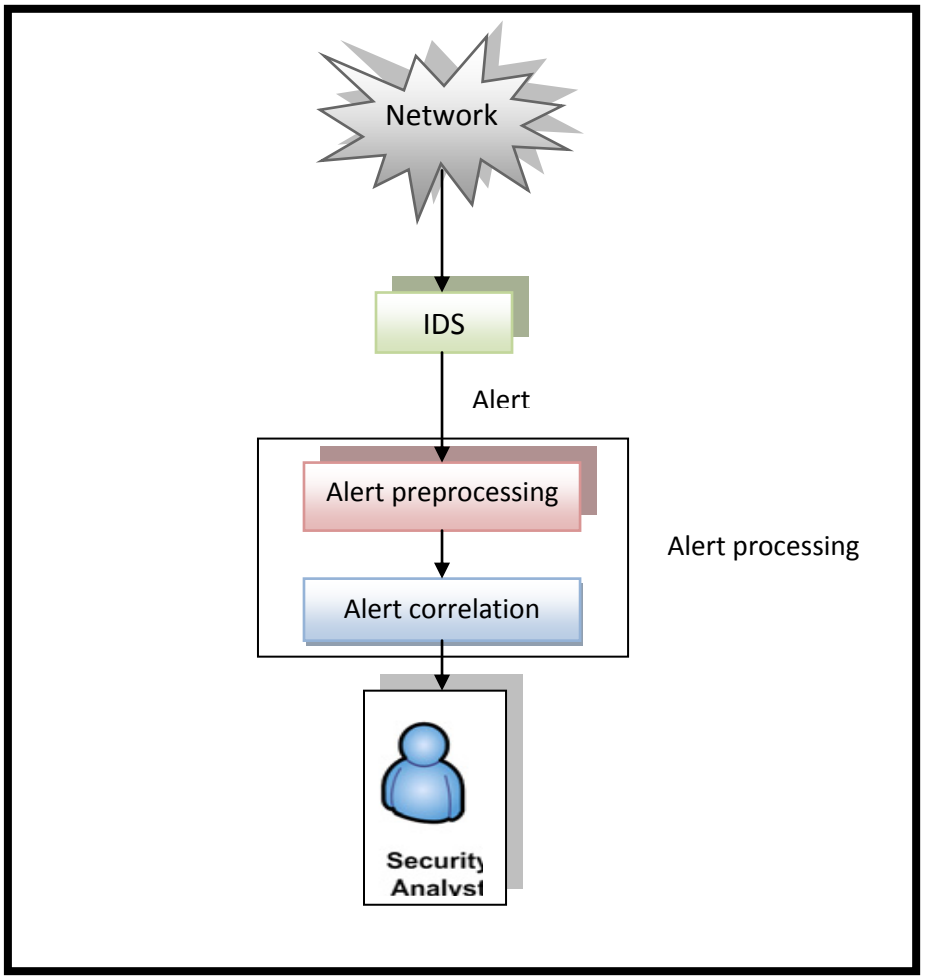


Figure (1): The Proposed System platform

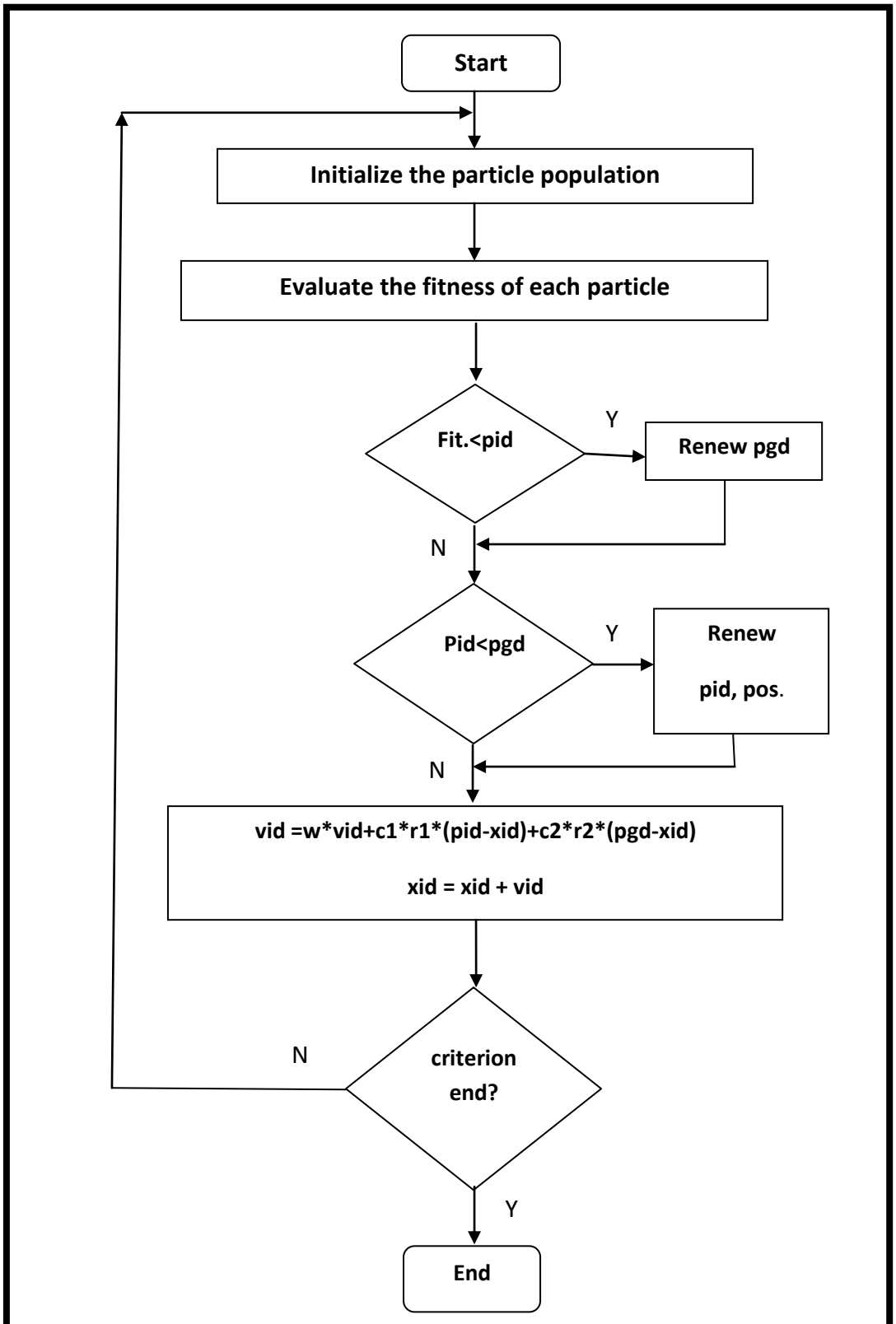


**figure 2: Flowchart of PSO Algorithm**

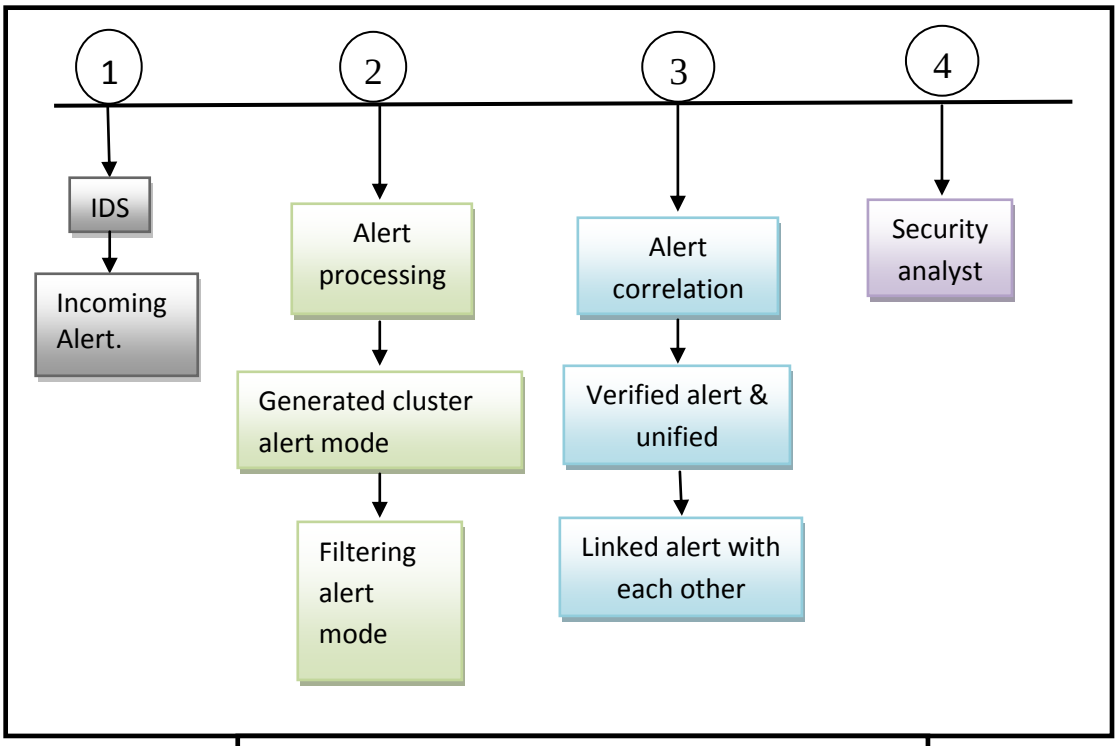


**Figure 3: described for system parts**

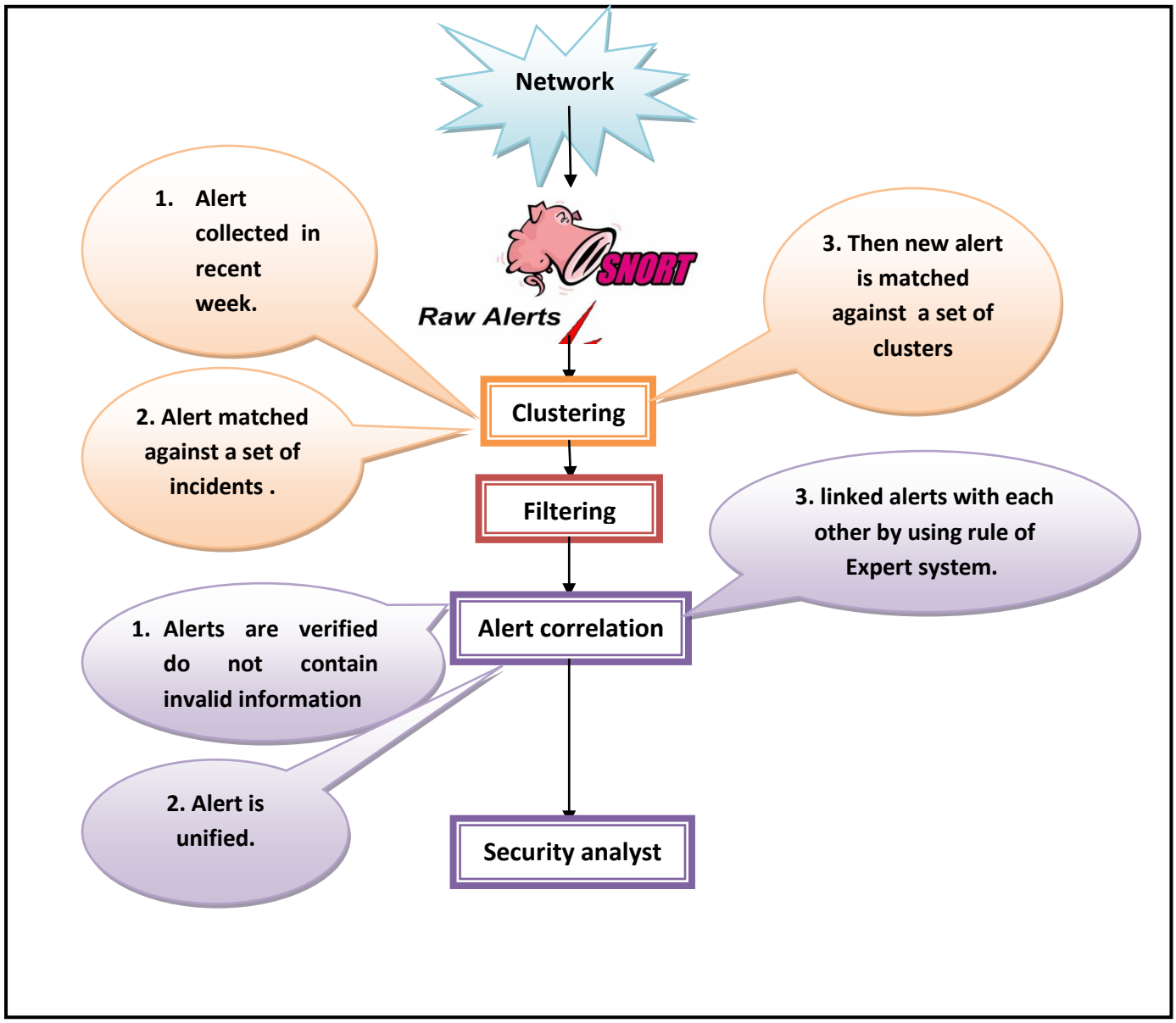


Figure 4: installation system