Protecting grid computing networks from cross-domain attacks using security alert sharing mechanisms and classification of administrative domains in security levels

Raheel Hassan Syed

▶ **To cite this version:**

Raheel Hassan Syed. Protecting grid computing networks from cross-domain attacks using security alert sharing mechanisms and classification of administrative domains in security levels. Cryptography and Security [cs.CR]. Université de Franche-Comté, 2012. English. NNT : 2012BESA2038 . tel-01622254

HAL Id: tel-01622254
https://theses.hal.science/tel-01622254

Submitted on 24 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE FRANCHE-COMTÉ
UFC/FEMTO-ST INSTITUTE
DÉPARTEMENT D'INFORMATIQUE DES SYSTÈMES COMPLEXES
(DISC)
UMR CNRS 6174
FRANCE

N° attribué par la bibliothèque

THÈSE

pour obtenir le grade de

DOCTEUR de l'Université de Franche-Comté

UFC/FEMTO-ST Institute

Spécialité: **Informatique**

présentée et soutenue publiquement

par

Syed Raheel Hassan

le 20 July 2012

Protecting Grid Computing Networks from Cross-Domain Attacks Using Security Alert Sharing Mechanisms and Classification of Administrative Domains in Security Levels

JURY

Président et Examinateur	Prof. Peter Kropf, Université de Neuchâtel, Switzerland.
Directeur de thèse	Prof. Julien Bourgeois, Université de Franche-Comté, France.
Rapporteur	Prof. Jianhua Ma, Hosei University, Tokyo Japan.
Rapporteur	Prof. Andrea Clematis, Consiglio Nazionale Delle Ricerche, Italy.
Examinateur	Dr. Abdoul Karim, President Efficient Protection Inc., Canada.

Dedication

To the orphan children and all those who suffer from hunger.

Acknowledgments

I am very thankful to the Higher Education Commission (HEC) which funded my studies and Quaid-e-Awam University of Engineering Science & Technology (QUEST), Pakistan who selected me for the scholarship to pursue PhD in France. I am thankful to the FEMTO-ST Institute, département d'informatique des systèmes complexes (DISC), for accepting me as a PhD student in their researcher group which is part of the Université de Franche-Comté (UFC) at Besançon in France. Throughout my tenure in lab the director of the thesis Professor Julien Bourgeois helped me in every step of my studies and administrative issues that I faced time to time. He is very kind, polite in nature and always encouraged me to work hard. He supervised my research by giving valuable tips from his years of experience and motivated me to collaborate with other researchers in different universities. He provided funding for going to attend the international conferences, summer schools and workshops in different countries. By his continues support I won the second position in the computing challenge using XtreemOS in Italy and did a successful collaborative work with the University of Emory in USA.

I am very thankful to my parents, brother and sisters who fully supported and encouraged me throughout my studies. I am thankful to all the members of the lab for their full support. I found Mouhannad Alattar a very helping lab-mate with whom I had very friendly discussions during the day. Abdou Wahabou who never said no to me and present when ever I need any help especially in French to English translations and helping me in the visits to the Doctors. Bogdan Cornea a very kind person with whom I had discussions on different subjects particularly in English. Matteo Cypriani the head of Belfort/Montbéliard association for the PhD students. He provided very active support in all the problems especially for resolving the administrative issues. Wassim Ramadan who gave me very useful tips especially when I had problems in operating systems, programming, networking, databases, Latex etc, and gave very technical advices. He helped me many times when I was under stress. Wassim gave me very useful lectures to learn French language. His lectures helped me a lot to improved my French language skills quickly. Kahina Boutoustous helped me in my early days when I came to France. She did a lot of administrative work for me which was required by a student who joins the university such as the residence, transport, banks, prefecture and student card. Jean-Laurent who helped me in reopening the telephone line at my residence and

provided Internet connectivity when I had disconnection problems during the week-ends. I visited many nearby cities of Montbéliard and cities in other regions of France with him.

I am thankful to Abdoul Ganame who helped me a lot to understand the DSOC and gave me many week-ends via connecting remotely from Canada in my machines at the lab. Renaud Bidou who provided some useful information by his emails and explained the basics concepts of core SOC model. His detailed and very useful documentation helped me a lot especially the work he has one in the area of network security.

I am thankful to the ASCAP for selecting me to play in team 1 and represent ASCAP in the inter-club tournaments of badminton in the region of Franche-Comte. I will never forget the time I passed with my badminton team members and friends Abdoul Hadi, Jawad and Thomas who provided a very friendly environment in Montbéliard.

Abstract

In recent years security is becoming a challenge in grid computing networks. Anti-virus softwares, firewalls and intrusion detection systems are not enough to prevent sophisticated attacks fabricated by multiple users especially when the number of nodes connected to the network are changing over the time. Grid computing networks are often composed of different administrative domains owned by different organizations, such networks are referred as multi-administrative domain networks. Each domain can have its own security policy and may not want to share its security data with less protected networks. It is therefore more complex to ensure the security of such networks and to protect them from cross-domain attacks. Due to the nature of grid computing networks, security pitfalls are plethora and adversaries are always sneaking to launch attacks. The main difficulty is to deal with the specificities of grid infrastructure, that are: multi-sites networks, multi-administrative domains, dynamic collaboration between nodes and sites, high number of nodes to manage, no clear view of the external networks and exchange of security information among different administrative domains. Grid computing networks aggregate huge computing power which they need for solving different scientific problems. This power can be used for attacking the grid's components as well as external networks. Attacks such as the Denial of Service (DoS) could be used to target user machines, servers and security management systems to sabotage the normal operations of the grid computing network. Attackers can use multiple nodes to launch distributed DoS attacks which generate large amount of security alerts in the network. On one hand this large number of security alerts degrades the overall performance of the network and creates instability in the operation of the security management systems. On the other hand they help in camouflaging other real attacks.

To handle all the above mentioned issues, in this thesis I am proposing a Security Event Manager (SEM) called Grid Security Operation Center (GSOC). GSOC can assist IT security managers in giving a view of the security of the whole grid network without compromising confidentiality of security data. To do so, GSOC provides a security evaluation of each administrative domain (AD) depending on the number of

security alerts reported. There are three security levels defined as level 1 is the most secure, level 2 is the more secure and level 3 is the least secure. The criticality of the security alerts is classified in three levels namely low, medium and high. Mathematical equations are proposed to assign each AD the respective security level. This classification helps to identify the ADs that are under attacks or the ADs that are at high risk of being attacked in future. The security evaluation also helps the resource sharing ADs to share their resources by imposing some restrictions, in case if there are high chances of the attacks in other member ADs of the grid computing network. A two step time based correlation mechanism is proposed which reduces the security alerts and continue detecting attacks under intense distributed attacks. At first step a Basic Correlation (BC) module is applied which tries to detect the attacks such as DoS and Brute Force (BF) locally within the same AD. The BC also minimizes the collected events from different computing elements in the network. If found any attack incident BC forwards the reported alert to the Advance Correlation (AC) module to further investigate the source of the attack. The AC is responsible for detecting distributed attacks such as DDoS within an AD. The BC and AC use one minute time window to detect and make sure that the reported alert is actually an attack. A parametric security alerts sharing scheme has been introduced. The SVOBox is the components added in the GSOC that does the security alert sharing. Security alerts can be shared at any time between the members of the grid computing network. This alert sharing informs the participating members to see the ongoing attacks on the other premises of the ADs without interfering in the security policy. This security alert sharing concept has been discussed in past but never implemented. GSOC is the first state of the art implementation of this idea. This alert sharing helps in blocking the propagation of cross-domain networks in grid computing networks.

The experiments are performed at the Grid'5000 network which is one of the biggest grid computing network in France and at the lab of UFC/FEMTO-ST. Multiple experiments are conducted and classified in four different categories. The first category tests the stability of the different security management systems. The second category shows the behavior of GSOC under different attacks. The third category explains the blocking of cross-domain attacks. The fourth category covers optimization in detecting distributed attacks. The thesis concludes by proposing the areas where the GSOC needs more improvements and followed with the future work.

Contents

List of Figures

List of Tables

Abbreviations

AC	AkabCollector
AC	Advance Correlation
AD	Administrative Domain
AM	AkabMaster
API	Application Programming interface
AR	AkabReport
AS	AkabSensors
AS-BM	AkabSensors Bandwidth Management
AS-IDS	AkabSensors Intrusion Detection
AS-LS	AkabSensors Log Server
AS-SA	AkabSensors Security Audit
AS-TM	AkabSensors Traffic Monitoring
BC	Basic Correlation
BF	Brute Force
EDG	European DataGrid
EGEE	enabling grid for e-sciences
FGIDS	Fault-tolerant Grid Intrusion Detection System
DN	Detection Nodes

DDoS	Distributed Denial of Service
DDS	Distributed Defense System
DIDSoG	Distributed Intrusion Detection System on Grid
DoS	Denial of Service
GIDS	Grid Intrusion Detection System
GSMS	Grid Security Management Systems
GSOC	Grid Security Operation Center
GUI	Graphical user Interface
LDIDS	Large-Scale Distributed Intrusion Detection System
IDSM	Intrusion Detection System Module
NC	network computing
NGS	National Grid Service
OCS	Open Computers and Software Inventory Next Generation
OSSIM	Open Source Security Information and Event Management
Pads	Passive Asset Detection System
PGIDS	Performance Grid Intrusion Detection System
PoD	Ping of Death
SANTA-G	Grid-enabled System Networks Trace Analysis
SEM	Security Event Management
SIM	Security Information Management
SIEMS	Security Information and Event Management Systems
SMS	Security Management Systems
RGMA	Relational Grid Monitoring Architecture

1

Introduction

1.1 Problem Statement

THE evolution and expansion of grid computing networks facilitated the scientists to utilize the shared computing power in a way that expedite the output of their research calculations. They have results in hours and minutes which took days and weeks in past. On one hand this emergence of computer networks provide many benefits to the research community. On the other hand this emergence of different networks inherits many security threats. These threats are changing their identity in the forms of different cybersecurity attacks. My motivation to work in this area is to identify the most critical and future threats that can interrupt the operation of grid computing networks. These threats can stop the operation of the network, in addition they can also infect and propagate to the networks that have very good security policies. Some of the possible threats are briefly discussed as under:

Due to the nature of grid computing networks, security pitfalls are plethora and adversaries are always sneaking to launch attacks. The main difficulty is to deal with the specificities of grid infrastructure, that are: multi-sites networks, multi-administrative domains, dynamic collaboration between nodes and sites, high number of nodes to manage, no clear view of the foreign networks and exchange of security information among different domains.

Grid computing networks aggregate huge computing power which they need for solving different scientific problems. This power can be used for attacking the grid's components as well as outside computers. Attacks such as the Denial of Service (DoS) could be used to target user machines, servers and on the security management systems to sabotage the normal operations of the grid computing network.

In recent years security is becoming a challenge in grid computing networks. Anti-virus softwares, firewalls and intrusion detection systems are not enough to prevent sophisticated attacks fabricated by multiple users especially when the number of nodes connected to the network are changing over the time. Attackers can use multiple nodes to launch DDoS attacks which generate large amount of security alerts. On the one hand this large number of security alerts degrades the overall performance of the network and creates instability in the operation of the security management systems. On the other hand they can help in camouflaging other real attacks.

In single administrative domain networks there is only one security policy which can be evaluated by the IT security manager thanks, to monitoring and reporting tools. Grid networks are often composed of different administrative domains owned by different organizations dispersed globally. Such networks are referred to as multi-administrative domain networks. Each domain might have its own security policy and may not want to share its security data with less-protected networks, making it more complex to ensure the security of such networks and protecting them from cross-domain attacks.

1.2 My Propositions to Handle These Issues

To handle all the above mentioned issues, in this thesis I am proposing a Security Event Manager (SEM) called Grid Security Operation Center (GSOC). GSOC can assist IT security managers in giving a view of the security of the whole grid network without compromising confidentiality of security data.

To do so, GSOC provides a security evaluation of each administrative domain (AD) depending on the number of security alerts reported. There are three security levels defined as level 1 is the most secure, level 2 is the more secure and level 3 is the least secure. The criticality of the security alerts is classified in three levels namely low, medium and high. Mathematical equations are proposed to assign each AD the respective security level. This classification helps to identify the ADs that are under attacks or the ADs that are at high risk of being attacked in future. The security evaluation also helps the resource sharing ADs to share their resources by imposing some restrictions, in case if there are high chances of the attacks in other member ADs of the grid computing network.

A two step time based correlation mechanism is proposed which reduces the security alerts and continue detecting attacks under intense distributed attacks. At first step a Basic Correlation (BC) module is applied which tries to detect the attacks such as DoS and Brute Force (BF) locally within the same AD. The BC also minimizes the collected events from different computing elements in the network. If found any attack incident BC forwards the reported alert to the Advance Correlation (AC) module to further investigate the source of the attack. The AC is responsible for detecting

distributed attacks such as DDoS within an AD. The BC and AC use one minute time window to detect and make sure that the reported alert is actually an attack.

A parametric security alerts sharing scheme has been introduced. The SVOBox is the components added in the GSOC that does the security alert sharing. Security alerts can be shared at any time between the members of the grid computing network. This alert sharing informs the participating members to see the ongoing attacks on the other premises of the ADs without interfering in the security policy. This security alert sharing concept has been discussed in past but never implemented. GSOC is the first state of the art implementation of this idea. This alert sharing helps in blocking the propagation of cross-domain networks in grid computing networks. Alert sharing can then be tuned in order to meet local security policy rules.

1.3 Structure of the Dissertation

The thesis is organized in five chapters which includes multiple sections, subsection, tables, figures and graphs. A brief introduction of the chapters are as follows:

Chapter 1 Introduction to the security issues in grid computing network and my proposition to solve these issues in this thesis.

Chapter 2 consists of the state of the art and related work with the comparative tables. In this chapter, I discussed some early definitions of the grid computing networks and how the application of grid computing networks changes with time. This chapter first covers the introduction of some of the major grid computing networks across the globe and their applications in the fields of science. A follow up with a brief explanation of some of the Security Information and Event Management Systems (SIEMS) with their architectural, attack detection and product comparison. This chapter also includes some Grid Security Management Systems (GSMS) with their internal architecture, drawbacks and their comparison with the core features followed by the evolution of GSOC. The last part of this chapter covers some statistics of networks attack trends with some general and specific security issues and my proposition to handle them in grid computing networks. The chapter ends by presenting four tables which summarizes the related work and gives a detailed comparison of different SIEMS with each other.

Chapter 3 presents the detail explanation of the Grid Security Operation Center (GSOC). Its internal architecture with all the components namely; EBox, CBox, LA, R-CBox, GA, GIDB, and SVOBox. The basic and advance correlation help to downsize the network utilization, database and processing power in the grid network. This two level correlation detects more complex attacks in real time. The mechanism of generating an alert which covers all the steps from collection of events by the CBox

till the alert arrives to the LA where the final analysis is performed is explained. The application of R-CBox which helps to calculate the delta. The value of delta determines if any of the CBox or R-CBox is under attack. The feature of sharing selected security alerts with the members of the grid has been discussed. Mathematical equations has been proposed for security evaluation of the administrative Domains. Two types of security evaluation is done in GSOC. Static evaluation which is calculated from OpenVAS, Saint, and Nessus. Dynamic evaluation which is performed using the proposed equations. The last part of the chapter presents the introduction of XtreemOS and its services. Some experiments are conducted to detect the services of the XtreemOS which are stopped by an unauthorized user. The results of the experiments show that GSOC can also be deployable in XtreemOS and handle its security issues.

Chapter 4 consists of the experiments which show four types of experiments. The first is based to testify the stability of different Security Management Systems (SMS) and their comparison with the GSOC. The second shows the behavior of each component of GSOC and how they process and detect the distributed attacks. The third is the explanation of an attack scenario where GSOC helps to block the propagation of attacks among other members of the grid. The fourth is the experiment which helps to reduce false positives and detect the attack at a very early stage. It covers different types of attacks such as Brute Force (BF), Distributed Denial of Service (DDoS) its different variations in the form of Ping of Death (PoD), TCP SYN, and spoofing attack such as Smurf attack. The motive of the experiments is to show that how the attacks are fabricated and propagate. Only the attacks are discussed that are less intensive in nature which does not harm the smooth operation of the network. There exist more complicated variations of these attacks which are very destructive in nature and have serious consequences. These variations are not been tested in order not to harm the operation of the network.

Chapter 5 is the conclusion which summarizes all the objectives that were set at the beginning of my thesis are achieved successfully. The future work section discusses about the future extension of my work and new dimensions that can be added in GSOC.

2

State of the Art

2.1 Introduction

THE industrial and scientific communities are always looking for more computational power. To achieve this goal researchers have studied multiple solutions for interconnecting organizations in order to share computational resources, which has given birth to grid computing networks. There exist multiple definitions of a grid computing network some of them are,

Leonard Kleinrock, in the 3^{rd} of July 1969 at UCLA Press Release, predicted about future networks, *"As of now, computer networks are still in their infancy, says Dr. Kleinrock, But as they grow up and become more sophisticated, we will probably see the spread of computer utilities, which, like present electric and telephone utilities, will service individual homes and offices across the country."*

Ian Foster defined grid computing as *"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."* [11]

To address social and policy issues he modified the definition and stated as, *Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations."* [12]

K. Krauter et al. [13] defined grid computing as, *"A distributed network computing (NC) system is a virtual computer formed by a networked set of heterogeneous machines that agree to share their local resources with each other. A grid is a very large scale, generalized distributed NC system that can scale to Internet-size environments with machines distributed across multiple organizations and administrative*

domains".

Franco Travostino et al. [14] defined grid computing in the book "Grid Networks:" as, *"The grid is a flexible, distributed, information technology environment that enables multiple service to be created with a significant degree of independence from the specific attributes of underlying support infrastructure."*

Frederic Magoules et al. [15] defined the grid computing in the book "Grid Resource Management:" as, *"A hardware and software infrastructure that provides transparent, dependable, pervasive and consistent access to large-scale distributed resources owned and shared by multiple administrative organizations in order to deliver support for a wide range of applications with the desired qualities of service. These applications can perform either high throughput computing, on-demand computing, data intensive computing, or collaborative computing."*

From all these definitions and statements, I extracted some properties that are important for security monitoring of grid computing environments. I therefore propose the following definition of a grid computing network as "a combination of multiple administrative domain (AD) where each AD consists of multiple local and remote sites. These ADs share resources such as storage, computation and services dynamically with each other."

As grid computing networks are complex networks, users need an interface to access its resources. This includes command line tools, web portals, different application interfaces or Graphical user Interface (GUI). Mostly a grid computing network is interfaced with its users through a middleware. There are many different grid middlewares and the most used are gLite [16], Unicore [17], ARC [18], and Unibus [19]. The largest grid service in Europe is EGEE (Enabling Grid for E-sciencE) project [20] which is based on gLite software. EGEE connects many local grids from different countries for example, Germany NGI-DE-Grid [21], Netherlands BIG-Grid [22], UK National Grid Service (NGS) [23], and Belgium BE-grid [24]. Some of the applications of grid computing are astronomy [25], biomedicine [26], climate [27], economics [28], earthquakes [29], fusion energy [30], diseases [31], Neuroscience [32], and volunteer computing [33]. A comprehensive and classified list of different grid projects is available at [34]. An AD is an entity which follows homogeneous security policy within all its local and remote sites. Whereas grid computing networks are composed of different administrative domains that are often located in different countries. They have different security policies and they must respect possibly different laws in each country. This heterogeneity arise issues in the security management of grid computing networks which are not taking into account by existing middlewares or security management softwares.

In order to minimize the cost and maximize the efficiency, large corporates and

research institutions have to collaborate with each other. There exists substantial security risks because they are having direct access to each others resources. This problem is highlighted by Ian Foster et al. [12]. One solution to this problem can be overcome by forming virtual organizations and defining certain common sharing agreements. However this solution does not guarantee enough security as the attackers could belong to one of the collaborating organizations. The best solution is to deploy security management solution which can keep monitoring the malicious activities within and outside the grid computing network.

This chapter presents different security monitoring and management systems, their proposed designs and deficiencies. I have classified them in two categories. The first are the Security Information and Event Management Systems (SIEMS) which are the general purpose security solutions. They are designed to be deployed in traditional computer networks. They can also be deployed in grid computing network, discussed in Section 2.2. The second are the Grid Security Monitoring Systems (GSMS) proposed specifically for grid computing networks discussed in Section 2.3. Section 2.4 describes the evolution of GSOC. Section 2.5 consists of statistics of recent attacks, general and specific problems, followed by propositions to improve the security in grid computer networks. At the end of this chapter a table is given which summarizes the related work and shortcomings which exists in different security management systems.

2.2 Security Information and Event Management Systems (SIEMS)

SIEMS emerged from Security Information Management (SIM) system and Security Event Management (SEM) system. SIM is an off-line security management system where the logs from the network elements are collected and stored at the central location for detecting malicious activities. Whereas SEM is an on-line security management system which continuously monitors the network to detect the malicious activities at real time. SIEMS are designed to received logs and events from heterogeneous sources. The collected information is mostly stored at the central place upon which different techniques are deployed to detect on-line and off-line attacks. The concept of SIEMS is still new and it is still in progress. SIEMS have a very comprehensive reporting system which comprises of very attractive charts and different forms of timely reports. There are two types of SIEMS available: open source and freely available ones and commercially available ones which can be pricey. In this section I will discuss both types of solutions for better understanding. In the end of this section, I will also provide different comparative tables that show pros and cons of selected environments.

2.2.1 Open Source Security Information and Event Management (OSSIM)

OSSIM is a product of Alien Vault Professional Corporation[1]. It is an open source freely available SIEM but some advance reporting capabilities are not freely available. It is a framework for the management of the security infrastructure of the large organizations. OSSIM is a collection of multiple security tools which are, (i) Snort (Network Intrusion Detection System)[2], a well known open source intrusion detection and prevention system. (ii) Ntop (Network and usage Monitor)[3], a freely available network traffic monitoring tool, which can detect network security violations. (iii) OpenVAS (Vulnerability Scanning)[4], an open source vulnerability scanner. (iv) P0f (Passive operating system)[5] which identifies the TCP/IP communication by using very sophisticated passive traffic fingerprinting mechanism. (v) Pads (Passive Asset Detection System)[6] which sniffs for hosts and services that are running in the network using rule based detection engine. (vi) Arpwatch (Ethernet/IP address parings monitor)[7], an Unix/Linux utility that tracks for Ethernet address pairings. (vii) OSSEC (Host Intrusion Detection System)[8], an open source tool for host based intrusion detection system. (viii) Osiris (Host integrity Monitoring)[9], an Unix service which monitors changes in the file system, kernel and the network configurations of the host systems. (ix) Nagios (Nagios Ain't Gonna Insist On Sainthood)[10], a system and network monitoring tool. (x) OCS (Open Computers and Software Inventory Next Generation)[11] which manages the IT assets of the organization. Using the combination of the above mentioned tools, OSSIM detects complex and distributed attacks.

Figure 2.1 is the general overview of the OSSIM which shows its components and their interconnection. Each sensor collects information from a wide range of tools in its premises and normalizes the events in the OSSIM formatting. These events are correlated by up to four levels of correlation for reducing false positives. The number of sensors can be increased depending on the density of the computing nodes that are to be monitored. These events are forwarded to the OSSIM server which consists of a SIM module and a database. The OSSIM server processes all the alerts received from different sensors and saves them as forensic evidences which may be required later to

[1] www.alienvault.com/community
[2] www.snort.org/
[3] www.ntop.org/
[4] www.openvas.org/
[5] http://lcamtuf.coredump.cx/p0f3/
[6] http://passive.sourceforge.net/
[7] http://linuxcommand.org/man_pages/arpwatch8.html
[8] www.ossec.net/
[9] http://orisis.shmoo.com/
[10] www.nagios.org/
[11] www.ocsinventory-ng.org/

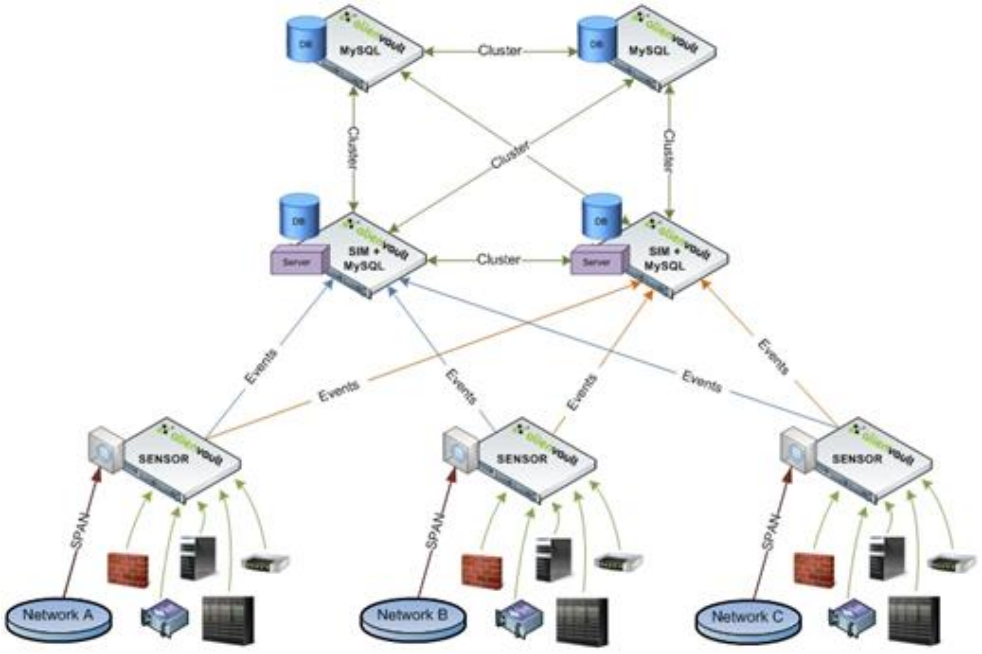


Figure 2.1: OSSIM Architectural Overview [1]

detect complex attacks. Depending on the size of the network and of the number of alerts received from different sensors the number of server modules can be increased. OSSIM has four levels of correlation. Level 1 passes all the events received from the sensors to the OSSIM server. Level 2 passes the events after counting the specified number of occurrences within a specified time period. Both the occurrence and specified time periods are customizable and are set by the administrator. Level 3 and 4 correlation are similarly the advanced versions of preceding levels of correlation. The details are available in the OSSIM documentation [1]. The OSSIM attack detecting, correlating and reporting system is very accurate and advanced but it fails in processing the alerts in real time especially under intense attacks. The main reason for this is that each event passes through multiple hierarchy of tools and every tool has its overhead to process, correlate and decide whether it is an attack or not. This whole process takes some time for the alert to be available for the administrator at the GUI. This time delay allows the attackers a fair chance to camouflage critical attacks by mingling with DDoS attacks.

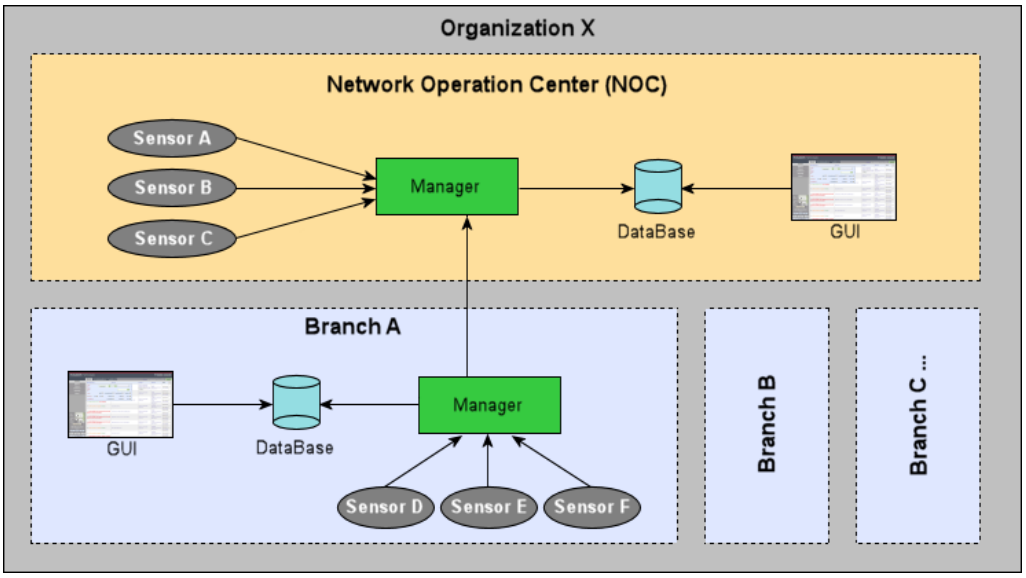


Figure 2.2: Prelude Architectural Overview [2]

2.2.2 Prelude

Prelude is a distributed Security Information Management (SIM) System[12] [2]. It is proprietary SIEM owed by prelude technologies, It is freely available for very limited operations. It composes of (1) Prelude-Manager, (2) Database and (3) Prewikka Interface as shown in figure 2.2.

(1) Prelude-Manager receives events from multiple sensors within local and remote branches of the organization. It can also receives logs from multiple Prelude-Managers which are distributed at different sites. The Prelude-Manager receives the events in the form of log files, databases or e-mails. Prelude-Manager is further composed of the following under given components. (i) **Libprelude** is an Application Programming interface (API) which provides a mechanism between the sensors and the Prelude-Managers to securely communicate with each other. (ii) **LibpreludeDB** is library which provides access to the Prelude database which contains all the reported alerts. (iii) **Prelude-LML** is a log analyzer that analysis the received events from different sensors and applications for suspicious behavior. (iv) **Prelude-Correlator** is an engine that correlates the alerts occurred between different Prelude-Managers. This is a rule based engine where the correlation rules are written to minimize false positives.

(2) Database which holds all the configurations and policies. It also stores all the reported vulnerabilities in the form charts and reports.

[12] www.prelude-technologies.com/en/welcome/index.html

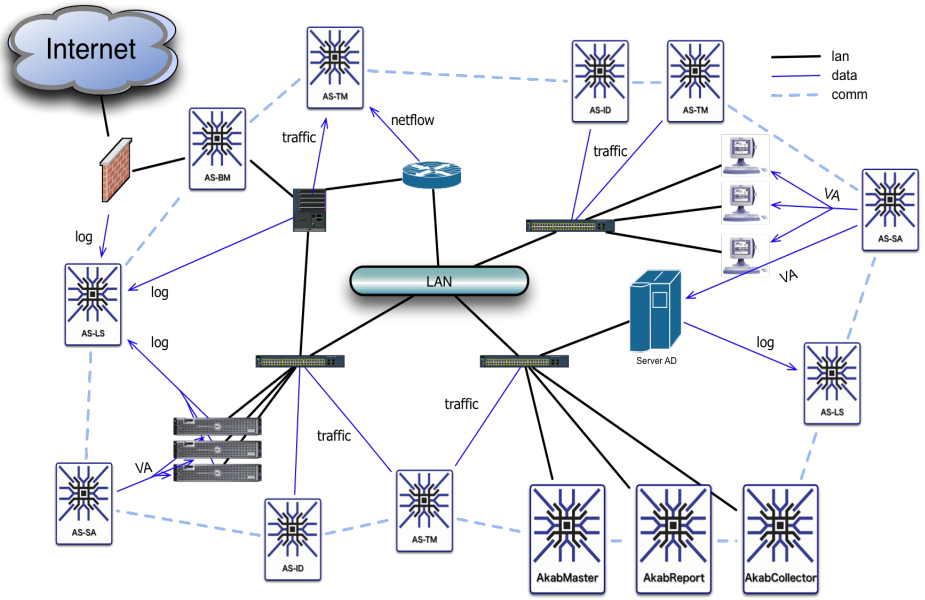


Figure 2.3: Akab Architectural Overview [3]

(3) Prewikka Interface is a Graphical User Interface (GUI) for Prelude SIM.

Besides sensor management and reporting alerts, it provides many features for example access to tools such as "whois" and "traceroute". Prelude SIM was under major changes since last two years. The prelude-technologies stop working on Prelude IDS in 2009 and now they resume in Jan 2012 by making some major changes. Recently they transform Prelude IDS into Prelude Pro 1.0 (SIM).

2.2.3 Akab

Akab is a SIEM system which has a scalable and modular architecture[13] [3] as shown in figure 2.3. It is a proprietary SIEM which is owed by Araknos, it is therefore not freely available even for short period of time. It is composed of AkabSensors (AS), AkabCollector (AC), AkabMaster (AM) and AkabReport (AR).

(i) **AS** is further composed of different network devices that are, Log Server (AS-LS), Intrusion Detection (AS-IDS), Traffic Monitoring (AS-TM), Bandwidth Management (AS-BM) and Security Audit (AS-SA). The **AS** collects events from the above mentioned devices and converts them into Akevent format which is a proprietary for-

[13] www.araknos.it/en/prodotti/architettura-akab.html

mat of the Akab SIEM.

(ii) **AC** collects the events sent from multiple AkabSensors. It then correlates the events to detect actual attacks in real-time. The real time and historical data allows Akab to provide a global view of the security of the entire organization. It has a multi-stage correlation architecture which uses rules, vulnerabilities or asset for reducing false positives. It has a time limit of 60 seconds for generating an alert if found any attack incidents in the collected events.

(iii) **AM** is a main security management system that manages all the components of Akab using web based GUI. Its console provides the mechanism for the configuration and the maintenance of the systems. It receives all the correlated events from the AC in the Akevent formate for more interpretation of the attacks that are still unknown.

(iv) **AR** contains database which store all the events for timely reporting feature. The reporting system can work in the absence of the administrator. If found some critical attacks, it forwards the alarm to the administrator using Mail, SNMP or Syslog.

2.3 Grid Security Management Systems (GSMS)

To monitor computational power, services status with their utilization and data storage there exist multiple solutions in grid computing networks as discussed in [35]. Figure 2.4 is a pictorial view which shows the composition of the grid computing network with its components. It presents the classification of different security monitoring and management systems with respect to computational, service and data grid networks. It highlights four levels where the security of the grid can be enhanced and the classification of the related work that has been done so far. It focuses the grid security level issues, for which I am proposing my solutions in this thesis. For security management in grid computing networks the solution should have all the basic attack detection capabilities as present in the traditional data networks. This means that a security management solution should detect the attacks at different levels: system, site and network level shown in figure 2.4. In addition, it must comply with the specific characteristics of the grid computing network namely: heterogeneity and dispersiveness of the computing elements and respect of multiple security policies. Furthermore, a GSMS must be scalable and fault-tolerant and it should share security alerts with the other members of the grid to increase it detection rate. As any other security management solution, it should resist to intense attacks like DDoS which can be done by a special correlation mechanism which can minimize the security alerts. It should detect distributed attacks and stop them propagating to other member organizations of the grid.

In this section, I will discuss some of the proposed grid security management solutions with their limitations in detecting attacks. In the end of this section, I will

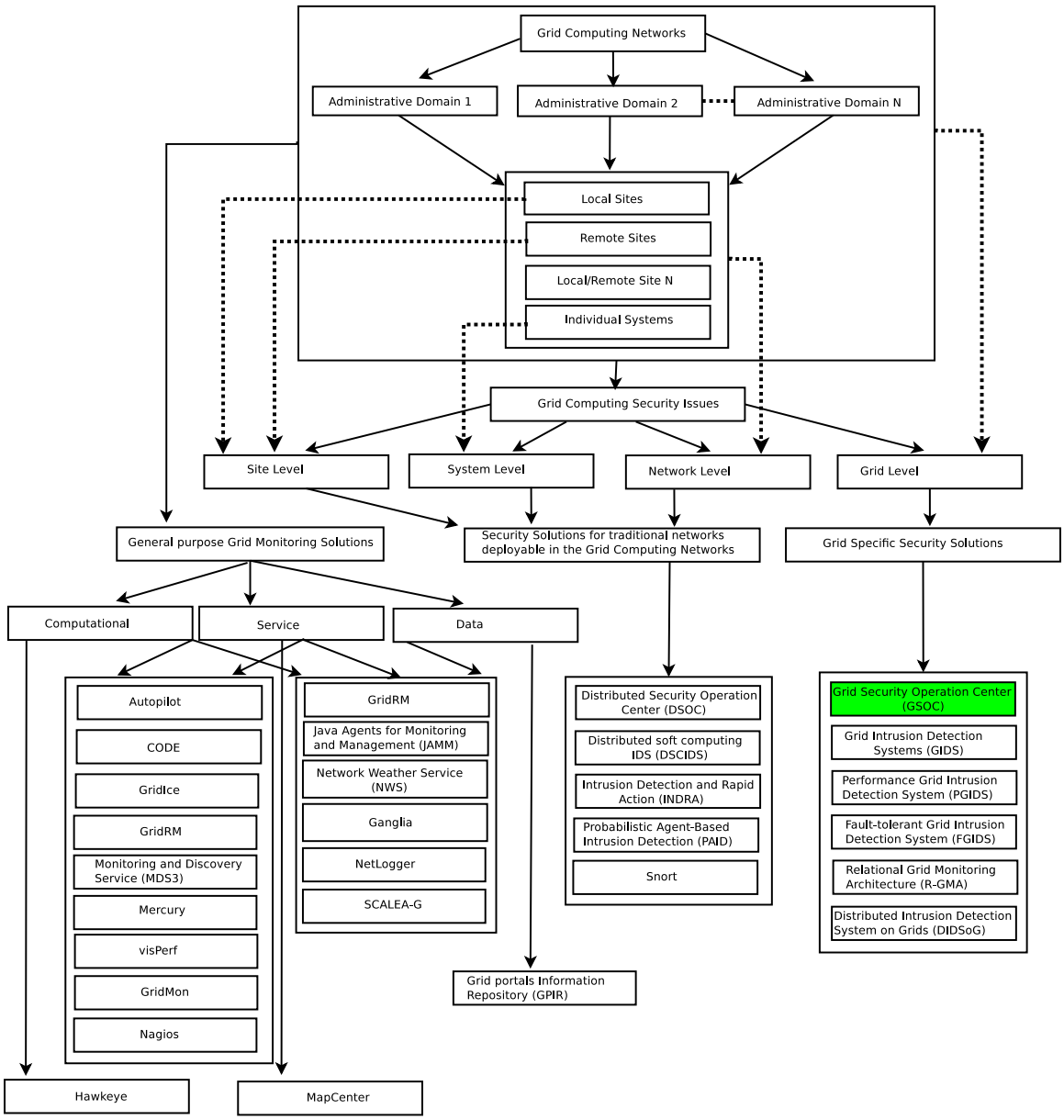


Figure 2.4: Classification of tools which monitor grid computing networks and security management tools for traditional and grid computing networks

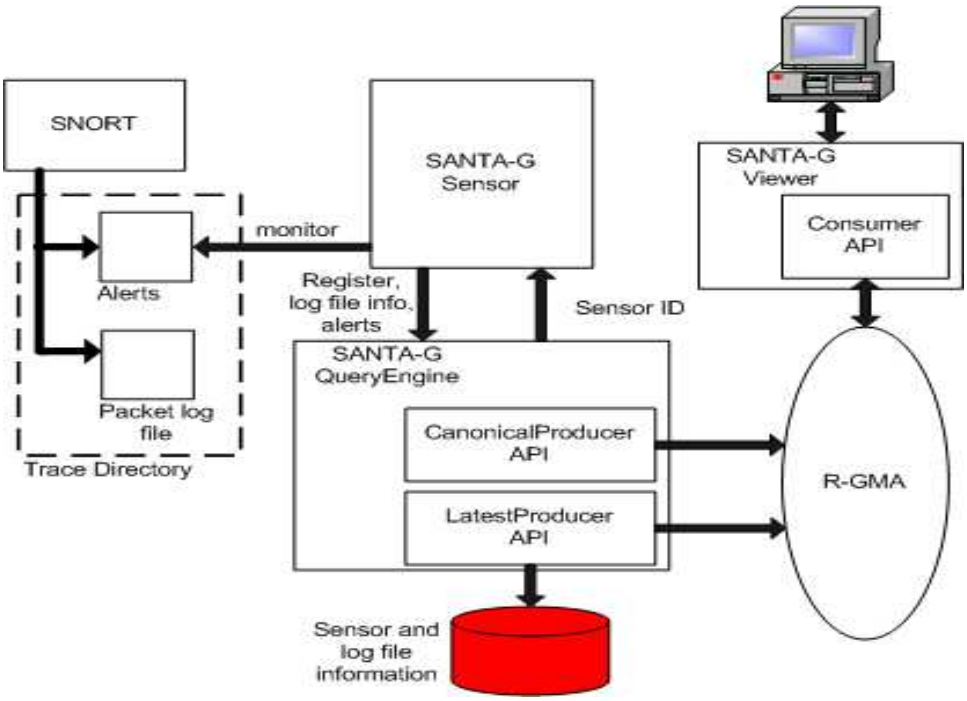


Figure 2.5: SANTA-G Architectural Overview [4]

provide a comparative table that shows the efficiencies and deficiencies present in the proposed GSMS.

2.3.1 Grid-enabled System Networks Trace Analysis (SANTA-G)

Kenny and Coghlan [4] proposed **SANTA-G** (Grid-Enabled System Networks Trace Analysis) which is based on the **RGMA** (Relational Grid Monitoring Architecture), is an implementation of **GMA** which is developed under the European DataGrid (EDG). SANTA-G uses Snort [36] for monitoring network traffic and is composed of three components: **Sensors** that need to be installed on the monitored devices, a **Query Engine**, and a **GUI** as shown in figure 2.5. Snort logs suspicious activities that occur in the network. These logs are then forwarded to a SANTA-G sensor which analyzes them and looks for attacks. If a new attack is found, the corresponding log will be sent to the query engine and saved in the database. The query engine publishes the detected attack to its users.

The SANTA-G model lacks incident detection a tracking and response platform, analysis of reported events to check the patterns for distributed attacks, and meaning it can not properly detect DDoS (Distributed Denial of Service attacks). The RGMA

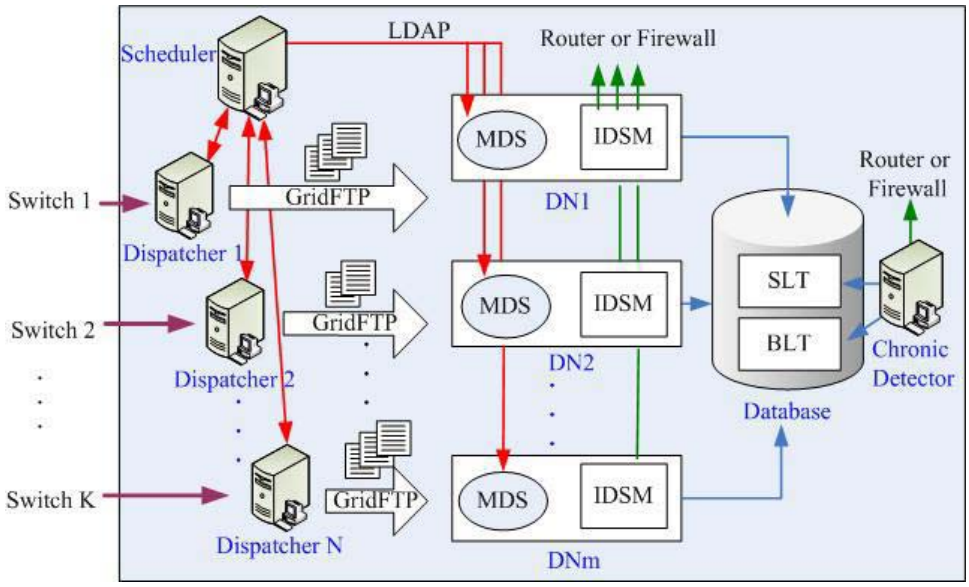


Figure 2.6: GIDS Architectural Overview [5]

consists of only one major database which stores all the configurations and attacks information sent by multiple SANTA-G sensors. When the size of the network grows rapidly and multiple SANTA-Gs start sending their alerts simultaneously. It creates overhead in the network and also consume a huge portion database in short period of time. Due to this design limitations, It is difficult to hold the alert information for long period of time. It can therefore, only correlate reported attacks for a short period of time. This lowers its detection capability for attacks or scans that are repeated with some time delays. SANTA-G only uses Snort as a source of data, giving a restricted view of the network security. SANTA-G does not have a security alert sharing mechanism which prevents it to detect cross-domain attacks.

2.3.2 Grid Intrusion Detection System (GIDS)

Fang-Yie Leu et al. proposed three versions of an intrusion detection system dedicated to grid networks: GIDS (Grid Intrusion Detection System), PGIDS (Performance GIDS) and FGIDS (Fault-tolerant GIDS) [5,37,38]. All the variations of GIDS consists of four types of components as shown in figure 2.6. (i) **Dispatchers** which assign network traffic to Detection Nodes (DN) for detecting attacks. (ii) **A scheduler** to balances the load between dispatchers. (iii) **DN** which use Intrusion Detection System Module (IDSM) for packet analysis and for detecting attacks. (iv) **Block List Database (BLD)** which holds intrusion information and suspected IP addresses.

The objective of GIDS is to detect three different kinds of attacks which are: logical, momentary, and chronic attacks. In GIDS, these attacks are defined as the following. Logical attacks start indirectly by the execution of any operation defined by the attacker, they last for a particular time duration and stop after a defined time period. Momentary attacks are launched by the attacker for short period of time, whereas chronic attacks last for very long period of time.

The accuracy of GIDS regarding attack detection is not very good because GIDS does not matches the patterns of similar attacks that have occurred in the past by the same attacker. The scope for attack detection is very small [5] because only TCP, UDP and ICMP flood attacks are used. To overcome these issues PGIDS has been proposed.

2.3.2.1 Performance Grid Intrusion Detection System (P-GIDS)

The objective of PGIDS is to add DoS/DDoS attacks detection to GIDS, but PGIDS suffers from DN failure under massive DDoS attacks. It uses Score Subtraction Algorithm (SSA) and Score Addition Algorithm (SAA) to improve the performance of DN. PGIDS is suitable for static environment where the computing elements are fixed but its detection capability is very limited in dynamic environments. To handle these problems a new version of PGIDS called FGIDS has been proposed.

2.3.2.2 Fault-tolerant Grid Intrusion Detection System (F-GIDS)

FGIDS introduces a new module called Backup Broker which helps the scheduler to assign another DN to a dispatcher if a massive attack occurs. FGIDS collects events from multiple sites of an administrative domain but as it has no correlation method for security alerts, it could be vulnerable to DDoS attacks that use grid computing power. Distributed attacks can be detected in one administrative domain but they cannot be detected if they target devices that are located in different administrative domains. More generally, cross-domain attacks cannot be detected by the different versions of GIDS.

2.3.3 Large-Scale Distributed Intrusion Detection System (LDIDS)

The architecture of LDIDS has been proposed by Yonggang et al. [39]. LDIDS is a scalable and fault-tolerant solution because each node works as a separate IDS. It is the combination of all these IDS that makes the LDIDS architecture. LDIDS has a graph structure which consists of a root node, many branches and leaf nodes. The leaf nodes are responsible for collecting the events from the local network.

The branch nodes are the intermediate nodes that monitors one or many child nodes. More specifically, branch nodes can be leaf nodes or next layer of branch nodes. The root node is the main node of LDIDS and it manages all the operations.

Every node has four layers namely (i) Lowest Layer, (ii) Pre-processing Layer, (iii) Analysis Layer and (iv) Harmonization and Management Layer.

(i) The lowest layer is responsible for receiving events from its child nodes.

(ii) The pre-processing layer normalizes the received events.

(iii) The analysis layer correlates the received events and detects intrusions.

(iv) The harmonization and management layer manages all the operations for detecting intrusions and provides a main point to interact with other components of LDIDS.

LDIDS can be applied in grid computing networks due to its modular nature but LDIDS lacks in the efficiency of the communication between its security components. Furthermore, in their research work [39], no details are given about the types of attacks that are detectable by LDIDS.

2.3.4 The Distributed Intrusion Detection System on Grid (DIDSoG)

DIDSoG, proposed by Poula Silva et al. [40], is a hierarchy of multiple intrusion detection systems. The services, discussed in their research work, are data gathering, data aggregation, data correlation, analysis, intrusion response and management. The sensors that are deployed in different parts of the network collects events from different hosts and applications. The collected events are sent to simple analyzers which try to detect intrusions by correlation and aggregation. This is called first level of complexity. If intrusions are found at the first level, they are then sent to the second level of complexity which performs a more complex analysis. If the intrusion is confirmed at the second level, the monitoring service is invoked to take countermeasure steps on the ongoing intrusion. The experiments are performed using a grid simulator called Gridsim which does not really reflect reality. Furthermore, no explanation is given on the communication between the IDS in order to detect intrusions. DIDSoG does not provide a mechanism for sharing alerts between the different administrative domains. Finally, even if the concepts of DIDSoG seems interesting, more development are needed before a deployment in a real environment.

2.3.5 Distributed Defense System (DDS)

Yang Xiang and Wanlei Zhou [41] proposed DDS for detecting and protecting grids from DDoS Attacks. Their system is based on statistical methods [42] to rapidly and accurately detect the intrusions. All the experiments which were conducted are

based on two assumptions. First, the network traffic is distributed equally within all the nodes and second, attacker uses a spoofed IP addresses for the attacks. The first assumption can lead to generate many false positives, because in grid computing network the network grows dynamically which results in abnormal network behavior all the time. The second assumption about the spoofed IP addresses is also not realistic because attackers does not always attack with spoofed IP addresses. The reason for this is, that most of the time, non-assigned IP address in the secured networks are blocked by default. This hurdle does not allow the attackers to use spoofed IP addresses for their attacks. Most of the time, the attackers try to get access to a machine of a legitimate user and from that machine they launch attacks which are harder to investigate. The attackers reserve many machines in the grid and pretend themselves to be a resource provider or as legitimate users but actually their intentions is to use these machines for launching different attacks. In grid computing networks, many machines can be reserved simultaneously. This gives attackers a good chance to use the computational power of grid on its own services and users.

For the experiments they have selected SSFNet (Scalable Simulation Framework) [43]. Their proposed distributed defense system requires access to the routers of each site. Therefore SSFNet provided this feature in order to capture and to analyze all the network traffic between different sites. The access to routers and the capture of network traffic of external sites are not possible in real active grid networks. The proposed system needs special permissions from other members of the grid. Therefore it is not clear whether their solution could detect high number of DDoS attacks in active grid network.

2.3.6 Grid-based Intrusion Detection System (G-bIDS)

G-bIDS is proposed by Choon and Samsudin [44]. It works as a backup of Grid Security Infrastructure (GSI), formerly called the Globus Security Infrastructure [45]. GSI provides secure communication between the components of a grid computational network using public key encryption (PKI), X.509 certificates and Secure Socket Layer (SSL) communication protocols. GSI also supports single sign-on for the users of the grid network. G-bIDS is composed of four main components (1) Agents (2) Server (3) Manager (4) Secure Communicator.

(1) **G-bIDS Agents**: are required to run on the monitored computing node. They are small programs that use minimal computing resources of the host node. The G-bIDS Agents are further divided in four sub components. (i) **The Communicator** which provides a secure communication channel with the G-bIDS server. (ii) **The Data Collector** which collects events form the monitored computing nodes, compresses and encrypts. This helps in minimizing the network bandwidth utilization. (iii)**The Re-**

sponse Engine responds in case of attack. (iv) **The Auditor** searches for the affected files.

(2) **G-bIDS Server**: is the main module that receives the events from the auditor presents in the G-bIDS agents. The G-bIDS Server is further divided in five sub components. (i) **The Communicator** which provides a secure communication channel with the G-bIDS agents. (ii) **The Analyzer** further verifies the events and try to find traces of the attacks. (iii) **The Policy Server** contains all the rules according to which the G-bIDS must work. (iv) **The Data Manager** does the management of resource utilization. (v) **The Service Engine** communicates with the G-bIDS manager for load balancing.

(3) **G-bIDS Manager**: is the controller that monitors and manages all the components and of the G-bIDS. The G-bIDS Manager is further divided in four sub components. (i) **The Policy Manager** which manages the local and all the other policies which are defined within and virtual organizations. (ii) **The Monitor** that monitors all the G-bIDS servers. (iii) **The Load Balancer** that manages load between all the G-bIDS servers. (iv) **The Console GUI** is for administrative purpose.

(4) **Secure Communicator**: securely provides availability and fault tolerance between the components of the G-bIDS.

G-bIDS can only monitor the sensors on which the G-bIDS agents are running. The deployment of the G-bIDS agents to monitor sensors in a grid network requires an intensive development task as there exist heterogeneous devices with different types of operating systems. G-bIDS cannot handle attacks which target the G-bIDS server and the G-bIDS manager. This shows that, when an attacker launches DDoS attacks on a G-bIDS server and manager, the whole G-bIDS operation could stop. Even if G-bIDS agents detect DDoS attack alerts, these alerts will not be processed and correlated by the G-bIDS server and manager.

2.3.7 Security for Grid Service

Von Welch et al. have worked [46] on modifications in Globus Toolkit version 2 (GT2) in order to upgrade it to Globus Toolkit version 3 (GT3). Their work introduces the first implementation of the Open Grid Service Architecture (OGSA). OGSA was first suggested by Ian Foster in his paper **The Philosophy of the Grid** [47]. OGSA is a type of architecture for service-oriented grid computing networks which has been developed in Global Grid Forum (GGF). OGSA provides heterogeneous systems with interoperability in order to communicate with different types of resources. The technical documentation of OGSA version 1.5 [48] recommends to use intrusion detection systems for handling DDoS attacks on grid services. OGSA does not provide any

mechanism to detect DDoS attacks from trusted users.

2.3.8 Predation and the Cost of Replication: New Approaches to Malware Prevention

Richard Ford et al. [49] used a ++shield program which is a modified version of the Shield program. The Shield program was developed by Wang et al. [50], it limits Malicious Mobile Code (MMC) in the network. In their experiments of the simulation of shield heuristic they used the improved version of the shield that was installed by default in all machines. If any machine is being attacked, the victim machine blocks the attack attempts by returning a magic number into the TCP headers or within the packet payload. This technique was useful to overcome Denial of Service (DoS) attacks but could not deal with DDoS attacks. In DDoS, the attackers use multiple sources with actual and spoofed IP addresses. Therefore, even if the attacked machine keeps blocking the requests, it cannot handle DDoS attacks.

2.4 Evolution of Grid Security Operation Center (GSOC)

GSOC is based on Common Intrusion Detection Framework (CIDF) proposed by Phil Porras et al. [51] and Stuart Staniford et al. [52]. CIDF has four main components namely Event Generators (EBox), Event Analyzers (ABox), Event Databases (DBox) and Response Units (RBox) as shown in figure 2.7. CIDF was the layered model which has three layers of services, (i) generalized intrusion detection objects (gidos) layer, (ii) message layer and (iii) negotiated transport layer. The Common Intrusion Specification Language (CISL) [53, 54] was planned to be used to securely communicate with the other components of the framework. The objective of CIDF workgroup was to develop a system that can be reusable and where the components of the framework must communicate with each other to detect attacks. The components can be added or removed by the security employees without shutting down the whole system. The components of the CIDF should look like one system for the external world but work in a distributed manner inside a network. The proposal of CIDF was very good but the project was stopped in 1999 without implementing the framework. From the available documentation [52], CIDF has some drawbacks. The intra-communication in the CIDF components has parsing problems due to the usage of CISL and three layered design which raises questions in the efficiency of the framework.

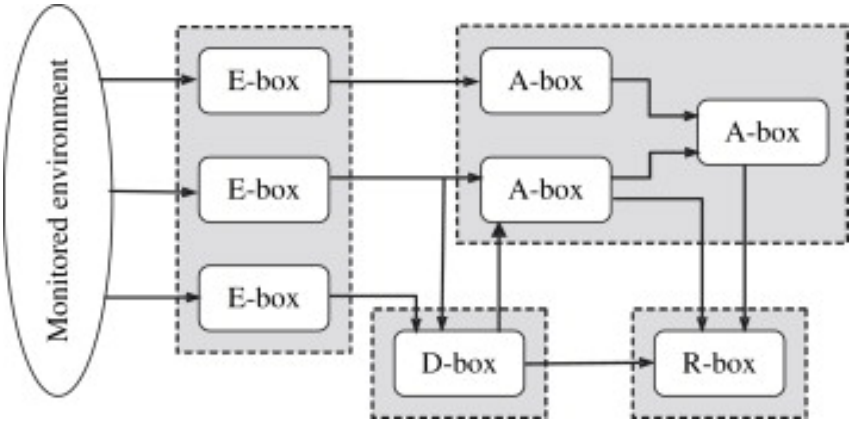


Figure 2.7: CIDF Architecture [6]

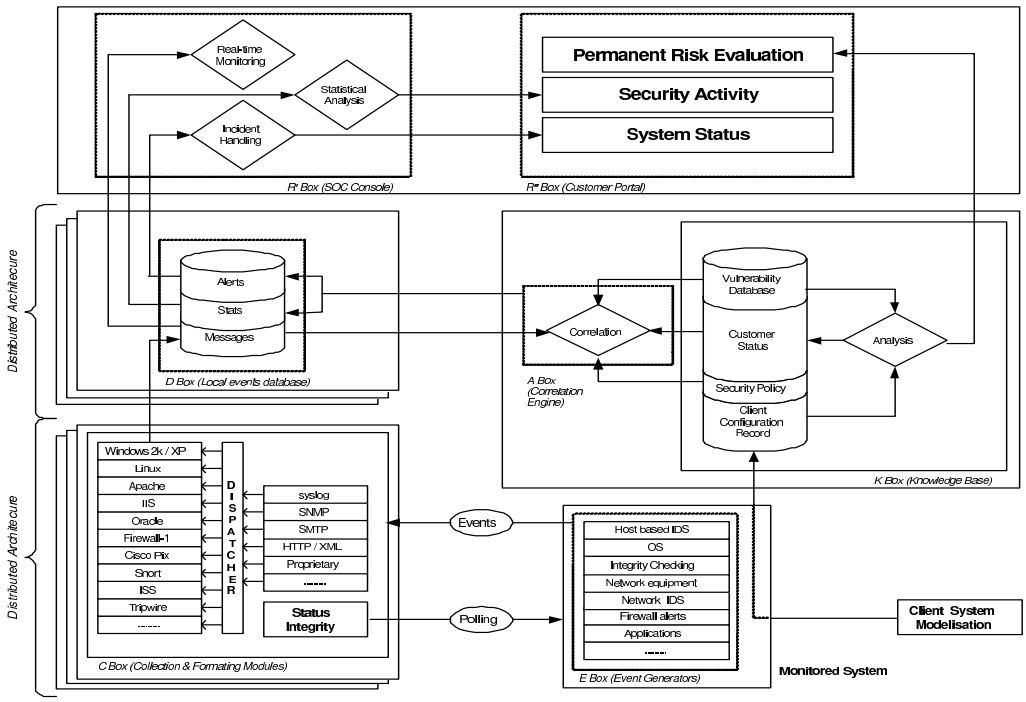


Figure 2.8: SOC Core Architecture [7]

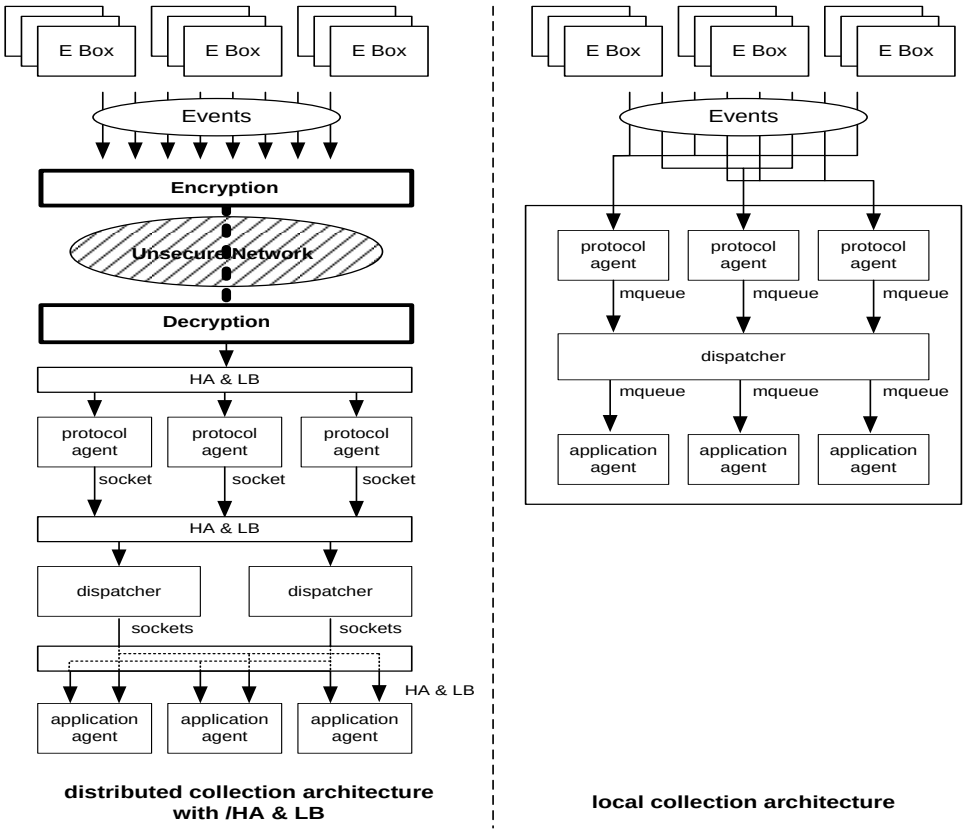


Figure 2.9: DSOC Core Architecture [8]

2.4.1 Security Operation Center (SOC) Box

The SOCBox proposed by Bidou et al. [7] has inferred the concept from CIDF [51, 52]. It is one of the earliest implementation of some of the core features of CIDF. The SOCBox has introduced one new box called Knowledge Base (KBox) in addition to the CIDF components. The core architecture of the SOCBox is shown in figure 2.8. The RBoxes were not implemented in SOCBox and a very basic correlation mechanism was introduced. The main idea behind SOCBox was to gather as much data as possible from every machine connected to the network so that the correlation can have more data to correlate. Although SOCBox shows great results in single site network but it fails when deployed in multi-site networks because it was based on centralized architecture. The main issues faced by the SOCBox are the database failure and its CBoxes stopped working under intense attacks.

2.4.2 Distributed Security Operation Center (DSOC)

DSOC was proposed by Ganame et al. [55]. In DSOC the internal architecture of SOC was modified to be able to deal with multi-site networks. Some new components have been introduced such as Local Analyzers (LA), Local Intrusion Database (LIDB), Global Analyzers (GA) and R-CBox. CBox were deployed across the network where they report to their corresponding LA. These LA then further interact with the LIDB for attack detection. In case of failures the master CBoxes were introduced. The R-CBox was implemented to ensure the response in case of the attack detection. The figure 2.9 has two parts, it represents the main design of DSOC . The right part represents the module which was used in the SOC for intra-communication of the components. Due to this module, SOC fails to work under multi-site networks because it cannot handle large requests coming from EBoxes. The left part of the figure shows the modified module which introduces encryption, High Availability (HA) and Load Balancing (LB) modules. These modules make the DSOC robust and scalable which worked under multi-site networks securely. The sockets were created to transport the events coming from CBoxes to the application agents via dispatcher. The DSOC gives encouraging results in multi-site networks and successfully detected the distributed attacks discussed in [55].

2.5 General Security Statistics and Their Classification

The Graphs 2.10 and 2.11 are taken from the Open Source Vulnerabilities Data Base (OSVDB) [9][14]. OSVDB is developed and maintained by the open source security community. The main objective is to provide unbiased, detailed and accurate technical information about the current vulnerabilities. The graphs show the attacks reported from 2004 till 2011 on different networks. It gives some statistics about the attacks trends on computer networks. The DoS attack is in the top six most launched attacks whereas other top three are Cross-site scripting (XSS), SQL injection and Cross-site request forgery (CSRF) attacks. The history and current status of the attacks show that attackers are continuously inventing new techniques despite of hardware and software solutions that protect computer network.

2.5.1 General Security Problems in Computer Networks

In recent years the expansion of computer networks has given birth to many security threats. These security threats, from minor to major fall in different categories. In the past, anti-virus softwares were used to protect the nodes locally, then firewalls came into being, which protect the nodes and network from outside attacks. But due

[14] http://osvdb.org/

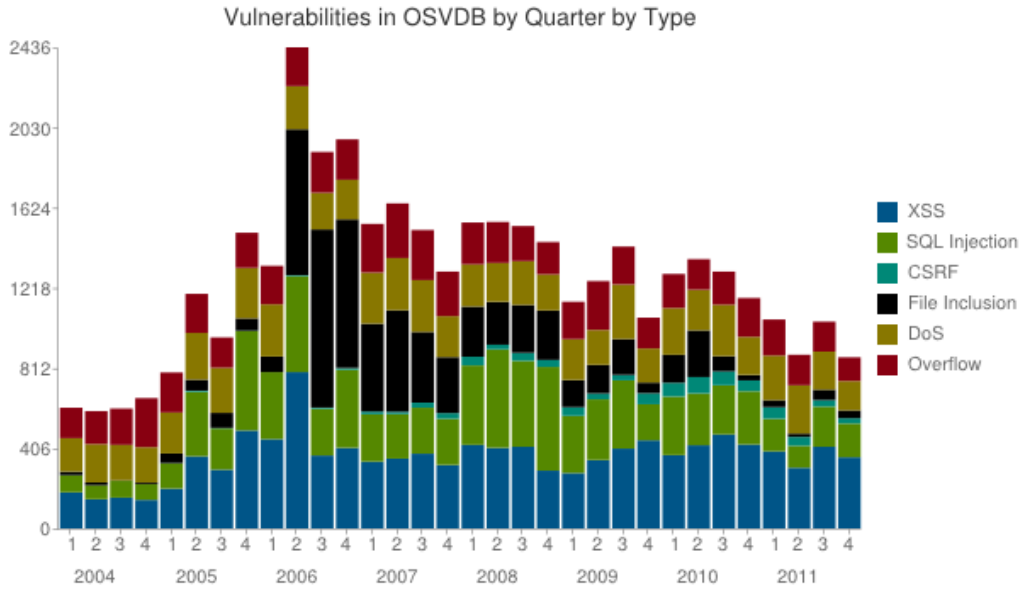


Figure 2.10: Histogram of the attacks reported during last eight years [9]

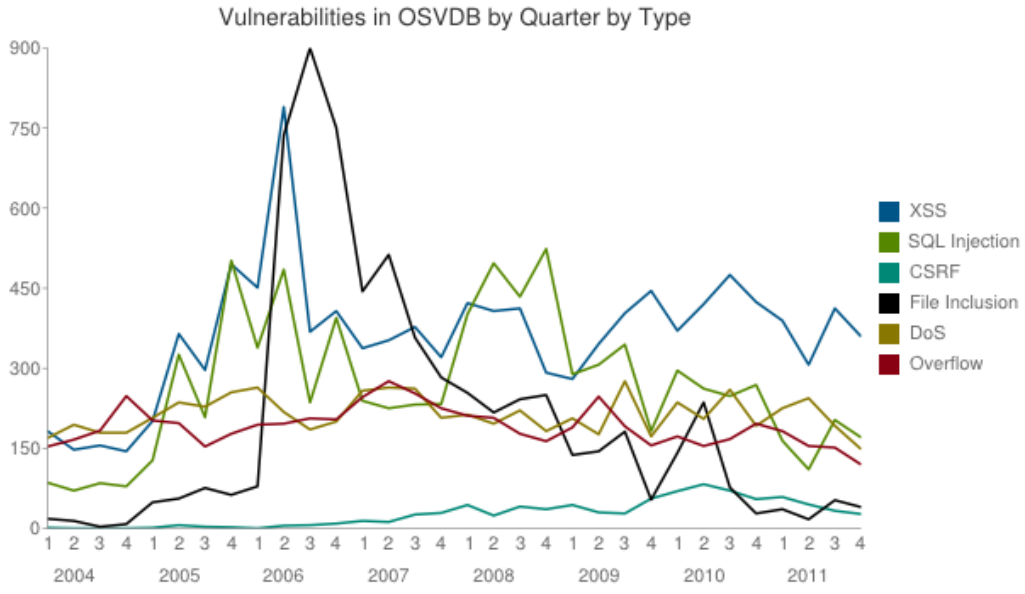


Figure 2.11: The trends of top six attacks in recent years [9]

to the continuous expansion in computer networks, intrusion detection and prevention systems (IDPSs) have been introduced. At present, IDPSs are also struggling to secure dynamic growing multi-administrative domain networks due to their performance limitations and the generation of too many false positives [56, 57].

In traditional computer networks, it is not recommended to send unencrypted passwords over the network as they can be easily sniffed out by the adversaries. Manually setting the passwords for large numbers of computing devices could lead to weak passwords, as automated tools are available which can crack the passwords very quickly using brute force attacks [58–60]. Communication using the symmetric-/shared secret-based authentication system is vulnerable by the man-in-the-middle attacks. The asymmetric-based authentication system is made vulnerable if the attackers use denial-of-service (DoS) attacks on the servers which maintains the certificates and the public/private keys. Most of the time the entire network is compromised from the users that use very simple passwords. Sometimes by the weird security administration that allows the attackers to gain access in the organizations network. The attackers also exploits the vulnerabilities that exist in the applications running in the network [61]. When one or multiple nodes are compromised in a single administrative domain network, it is easy to take quick actions on the hosts and network of the organization to identify the source of the problem. Once the source is identified, new policies and restrictions can be placed within the organization's network to block future threats.

2.5.2 Specific Security Problems in Grid Computing

For attackers, grid services are very interesting targets to violate quality of service (QoS) by launching DoS and distributed denial-of-service (DDoS) attacks. Section 6.4 of the RFC 3820 [62] mentions there are possibilities for launching DoS attacks on the machines that are responsible for generating key pairs and when granting dynamic delegations using proxy certificates. By the growth of web service and XML technologies in grid computing networks, the application level firewalls are unable to detect sophisticated attacks fabricated using content of the messages [63]. When a node in the grid computing gets compromised it is very hard to identify the source of the problem because there are multiple nodes from different administrative domains collaborating with each other. In such cases there is always a high possibility that attacks could be propagated to another organization's network which is the part of that grid network.

When an attacker launches an intensive attack on a grid computing network the IDS starts generating many security alerts. It starts sending these alerts to the central database. These huge number of security alerts create bottlenecks in the network and use a lot of disk space. Due to these intensive attacks, the IDS can become overloaded and turn unstable. The instability of an IDS results in generating security alerts which

are false positives or in generating too many alerts. Hence the security management system needs to do time consuming efforts to analyze these alerts which give the attackers a fair chance to perform malicious activities. The instability of an IDS stems from multiple reasons. The most common ones observed are due to disk space failure, database failure, system process queue overloading, excessive memory and processing power consumption. The same is applicable in grid computing network but the intensity of the attacks is much higher because the attackers have the choice to reserve as many machines they require. Thus they can launch more intensive attacks in less time period compared to the traditional networks.

2.5.3 Propositions for Improving the Security of Grid Computing Networks

Grid computing networks were invented to share computational resources from locations dispersed from all over the globe. It is a network which can be called as multi-administrative or multi-organization network. This emergence of different organizations has made the grid computing network vulnerable to more network attacks. Due to the nature of the grid, an attacker can use the grid computational power to target the resources of any administrative domain attached to the grid network. In these circumstances one possible threat are DDoS attacks which can halt the overall operation of grid computing network. There is therefore a high need to put a mechanism in place which can detect these attacks as early as possible while continuing operating stably. It should correlate the events before generating an alert. The correlation should minimize the alerts and generate one concise alert in order to inform the administrator. This correlation minimizes the network overhead and saves a lot of processing power and disk space of the sensors which are responsible for detecting malicious activities. There should be a mechanism which can evaluate the security of the sites and there should be security alert sharing between different sites.

IT managers always keep in mind that 100 percent security is an unrealistic objective [64]. To maintain the security up to maximum, grid computing networks possess Grid Security Infrastructure (GSI) [45] and Public Key Infrastructure (PKI) [65] that uses certificates for validating the legitimate users into the network. However, in [66] Cody et al. envision future research in grid computing will focus on high performance vs high security in grid computing networks because data encryption is inversely proportional to performance. In [67], Schwiegelshohn et al. quoted the example of the XtreemOS [68] project for native Linux system-level support for authentication mechanisms. They emphasize the need of mechanisms that reduce the load on middle-ware for security measures that can be shifted to operating systems. Propagation of cross-domain attacks can be blocked if the security information can be shared among the members of the grid computing network [69].

Despite all precautions and propositions, chances still exist that adversaries can target the victim whenever they receive the opportunity. Therefore, there is a high need to have a security monitoring system in place that works in parallel with other security components. It must be scalable and fault-tolerant. It must handle sophisticated network attacks launched using the power of grid networks, must can block cross-domain attacks, must report security breaches in real time, and must share them with other members of the grid computing network. In this thesis, I am proposing a security monitoring system that handles all the above mentioned propositions.

Table 2.1 shows a comparative analysis of some of the above discussed security management systems. Based on my knowledge I have selected the most important parameters for comparison. These parameters are the very basic and essential part of a systems that must be incorporated in a security system which manages the security of a grid computing network. The security solutions are categorized in four categories which are fully suitable, suitable, partially suitable and not suitable. The table summarizes the main features of the proposed solutions which shows that GSOC (Grid Security Operation Center) overcomes the limitations of the proposed solutions for grid computing networks. Table 2.2, 2.3, and 2.4 show the comparison of different SIEM systems. The comparison is based on their attack detection capability, architecture overview, and product features.

Table 2.1: Comparison of Different Grid Security Management Solutions

GSS	DoS	DDoS	BF	SAS	SE	Cor	CDA	SC	FT	Comments
GSOC	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Fully Suitable
OSSIM	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Suitable
Prelude	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Suitable
DSOC	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Not Suitable
R-GMA	Yes	No	Yes	No	No	No	No	No	No	Not Suitable
GIDS	Yes	Yes	No	No	No	No	No	No	No	Not Suitable
PGIDS	Yes	Yes	No	No	No	No	No	Yes	No	Not Suitable
FGIDS	Yes	Yes	No	No	No	No	No	Yes	Yes	Partially Suitable
LDIDS	Yes	?	?	Yes	No	Yes	Yes	Yes	Yes	Not Suitable
DIDSoG	Yes	No	?	Yes	No	Yes	Yes	Yes	Yes	Partially Suitable
DDS	Yes	Yes	?	No	No	No	No	Yes	No	Not Suitable
G-bIDS	Yes	?	Yes	Yes	No	No	?	Yes	Yes	Partially Suitable

GSS=Grid Security Solution, DoS=Denial of Service Attacks, DDoS=Distributed Denial of Service Attacks, BF=Brute Force Attacks, CDA=Cross Domain Attacks, SAS=Security Alert Sharing, SE=Security Evaluation, Cor=Correlation, CDA=Cross Domain Attacks, SC=Scalability, FT=Fault Tolerance, ? = Not known.

Table 2.2: Architectural Comparison of Different Security Information and Event Management Systems

Features	**OSSIM**	**Prelude**	**Akab**	**DSOC**
Open Source	Yes	No	No	No
Distributed	Yes	Yes	No	Yes
Scalable	Yes	Yes	Yes	Yes
Fault-Tolerant	Yes	No	No	Yes
Agent Based	No	Yes	Yes	No
Signature Based	Yes	Yes	?	Yes
Anomaly Based	Yes	?	Yes	No
Monitor Heterogeneous Devices	Yes	Yes	Yes	Yes
Manages Heterogeneous Data Sources	Yes	Yes	Yes	Yes
Programs installation required on Monitored Hosts	No	Yes	No	No

Table 2.3: Attack Detection Comparison of Different Security Information and Event Management Systems

Features	**OSSIM**	**Prelude**	**Akab**	**DSOC**
Single Stage Correlation	Yes	Yes	Yes	Yes
Multi-Stage Correlation	Yes	?	Yes	No
Rule Based Correlation	Yes	Yes	?	No
Security Evaluation	Yes	Yes	No	No
Cross Domain Attack Detection	No	No	No	No
Real Time Attack Detection	Yes	Yes	Yes	Yes
Security Alert Minimization	Yes	No	No	No
Security information sharing within Administrative Domain	Yes	Yes	Yes	Yes
Security information sharing intra Administrative Domain	No	No	No	No

Table 2.4: Product Comparison of Different Security Information and Event Management Systems

Features	**OSSIM**	**Prelude**	**Akab**	**DSOC**
Commercial	Yes	Yes	Yes	No
Forensic Events Logs	Yes	Yes	Yes	No
Current Development Status	Active	Not Active	Active	Not Active
Active and Technically Supported	Yes	No	Yes	No
Easily Configurable and Deployable	Yes	No	No	Yes

3

GSOC Architecture with its Features

3.1 Introduction

THE objective of my work is to develop a security operation center dedicated to multi-administrative domain networks such as grid computing networks. This chapter presents in detail the proposed design which monitors the security in the grid computing networks. It explains the internal architecture and interconnection of each component with its functionalities. The proposed design has two levels of correlation namely, basic and advanced which help to detect sophisticated and distributed attacks. This two-step correlation detect attacks while utilizing the minimum network resources such as processing power for event analysis, consumes less bandwidth because the duplicate events of the same attack are discarded at each local site of the network. Only those events are saved and processed which contains a high possibility of attack. They are further analyzed at the second step to make sure that it is going to detect the real attack. This first step takes 60 seconds while the next step further waits for 60 seconds and then reports about the attacks. This results in saving a lot of disk space because now it is not mandatory to store large log files and moving them within a network as a backup. Although a very detailed analysis of different alert correlation techniques has been discussed by Sadoddin [57], whereas the proposed solution only correlates the same types of attack incidents repeated in a specified time period. This time based correlation keep its modules stable under massive DDoS attacks. The security evaluation is performed internally which is done by itself, and another evaluation is performed externally by the tools available commercially. Both the security evaluations are combined together to verify the vulnerabilities in order to classify the network. The last part contains the brief introduction to XtreemOS with some of its security issues. More details are covered in the remaining part of the chapter. Section 3.2 presents the GSOC design. Section 3.3 explains the Basic and Advance correlation. Section 3.4 shows the

security alert generating mechanism. Section 3.5 is the detailed explanation of inter-communication of the components which are the part of GSOC. Section 3.6 presents the dynamic and static security evaluation adopted by GSOC. Section 3.7 is the introduction of XtreemOS with some experiments to monitor the services restarted by unauthorized users.

3.1.1 Shortcomings in the Grid Security Management Systems (GSMS)

The common problems which are observed in majority of the earlier mentioned security management systems are listed as,

(i) No proper mechanism was present to handle the attacks if the attacker targets its own components.

(ii) Except OSSIM and Prelude, correlation mechanisms were not very accurate, either they generate false positives or false negatives.

(iii) Except OSSIM and Prelude, no security sharing mechanism was present, therefore they fail to stop cross-domain attacks in multi-administrative networks like grid computing network.

(iv) Except OSSIM, security evaluation methods were proposed but not fully implemented in the systems.

(v) No method was present to calculate the deviation in the security alerts. Deviation helps in detecting DoS attacks in any local site of an AD.

(vi) During DDoS attacks they need much more disk space, processing power and network bandwidth.

(vii) There is no mechanism existing in order to know the type of attacks that are in progress at the external networks with whom they are sharing their resources.

Grid Security Operation Center (GSOC) proposes improvements in all the above mention shortcomings, which are discussed in detail in section 3.2. The experiments are performed in chapter 4, which validates that GSOC has shown improved results as compared to other GSMS.

3.1.2 Specific Properties of Grid Computing Networks

In addition to the above mentioned shortcomings, GSMS also failed to handle the specific requirements of grid computing network which are mentioned below:

- The grid computing network is a combination of different Administrative Domains (ADs), each of them composed of multi-site networks. Each AD follows

its own security policy. If the security policy of any one AD is broken by the attackers, this could have very serious implication on the other participating ADs.

- In the grid computing network a high number of nodes collaborate with one another. The size of the network is increasing and decreasing rapidly. This makes it dynamic in nature. The security system that manages the security of the grid network must have scalability that can handle a very large number of computing elements.

- In the grid computing network a view of the security events of other ADs is unavailable. The security system should have a security alert sharing mechanism which must provide the information of the reported attacks to the participating members of the grid.

Keeping the above mentioned issues in view, Grid Security Operation Center (GSOC) is introduced in this thesis. The GSOC has been proposed in [69, 70] and has shown improvements in the shortcomings faced by the GSMS. It incorporates all the grid specific properties while keeps the grid network secure. To some extent, the GSOC helps to cover two types of DoS attack solutions (i) the preventive solution and (ii) the reactive solution mentioned in [64]. The GSOC has been proposed because of its modular design. It has the scalability to monitor the security of the very large networks that are growing at an unpredictable rate. GSOC is easy to deploy and has distributed nature therefore it is most suitable for grid computing networks. It provides a global view of the security of the entire grid which makes it unique as compared to others. The global view of the security helps GSOC to detect cross domain attacks. It has a fault-tolerance architecture, which helps it to continue detecting the attacks even if the attacks are targeted on its own modules. GSOC does static and dynamic security evaluation to classify the ADs according to the level of security. This evaluation helps to determine the ADs that are under attack or likely to be attacked.

3.2 Grid Security Operation Center (GSOC) Design

GSOC is composed of seven components namely Event-Generating Box (EBox), Logs Collecting Box (CBox), Local Analyzer (LA(DBox+ABox)), Remote Logs Collecting Box (R-CBox), Global Analyzer (GA), Global Intrusion Database (GIDB) and Secure Virtual Organization Box (SVOBox). The details of the internal architecture of each components are discussed below.

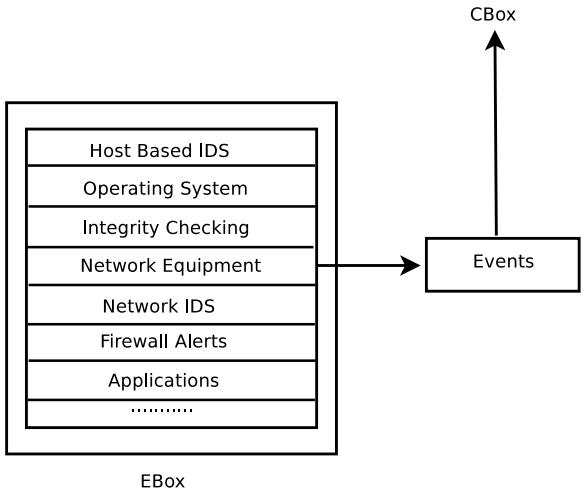


Figure 3.1: EBox Design

3.2.1 Event-Generating Box (EBox)

The EBox is a component in the grid network that generates events (figure 3.1). These events could be of two types. The first type of events come from the logs generated by connected equipments that exist in the network. These equipments could be firewalls, routers, switches, wireless hubs, or RADIUS servers. Mostly their operating systems collects the triggered events which are stored locally in these equipments. These events are then forwarded to the CBoxes that are present at few specific places in the network for analysis. The second type includes different kinds of third-party applications that generate events. The latter generates events when a specific state or a threshold value occurs in different network management systems (NMSs). These NMSs are very useful for detecting distributed denial of service attacks by continuous checking system availability via ICMP or SNMP [7]. If the service stops responding, they generate events defined by the administrators. These events contain raw information and they could have different formats depending on the application. One example of Syslog is,
Dec 6 16:06:05 192.168.8.65 [LOG_INFO] sshd[299]: Failed password for ROOT from 192.168.7.10 port 13322 ssh2.
The Syslog message shows the time, date, type of the incident, who tried to perform the operation, user names and IP addresses. This information is very useful to detect the attacker who continuously performs malicious activities in the network. When the attacker tries to launch attack, several abnormal messages are generated and saved in the logs. These type of events are then forwarded to the CBox.

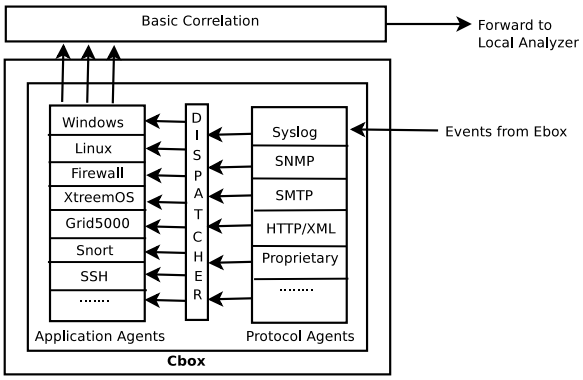


Figure 3.2: CBox Design

3.2.2 Collecting Box (CBox)

The CBox is a log-collecting module that collects logs from different EBoxes. One CBox is enough for one local site of an administrative domain. More than one CBox can be deployed in one site, if the number of generated events is too high. There are mainly two causes in which EBoxes generate high number of events: (i) At any local site some very sensitive machines are running therefore the administrator wants to log everything. This includes that EBoxes are configured to log every packet such as information, warning, notice, error, critical, emergency and alert messages. (ii) The second possibility occurs when one or multiple EBoxes are under attack and every time when the attacker tries to perform malicious attempts. The effected EBoxes logs the attempts and forwards them to the CBox. Every EBox has a different format for reporting the event.

The CBox itself contains three different sub modules: protocol agents, dispatcher and application agents. The CBox collects raw information from different protocols agents such as Syslogs, SNMP, SMTP shown in figure 3.2. The protocol agents transports these events from EBoxes to the CBox. The dispatcher plays an intermediary role between protocol agents and application agents. Application agents are the modules in the CBox which contain the possible attack lists. The dispatcher searches for these reported events from the EBox and tries to match them with available application agent modules such as vulnerabilities present in Linux, Windows and XtreemOS. When the reported event matches with any defined attack template, it is then arranged in an internal format before it is sent to the Local Analyzer (LA). The LA contains an alert reporting (ABox) and a database box (DBox) which resides physically in a different machine in the same network site or remote network site of an administrative domain.

The GSOC has correlation at two levels, the one at the CBox level which is called the basic correlation (BC) and the other one at the LA level which is called the

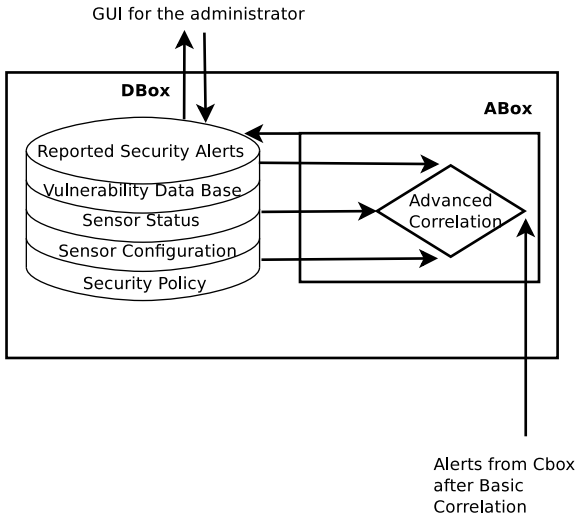


Figure 3.3: DBox and ABox Design

advanced correlation (AC). Their details are to be found in section 3.3. In the BC, the reported events are saved for a small period of time (approximately one minute) locally on the machine where the CBox is running. Afterwards, events that do not have the attack incidents are dropped and only the correlated alerts are stored in the local database and are forwarded to the LA. The correlation engine reports whether any of the events reaches a definite threshold and names it a weak brute force attack, a weak ping of death attack or a weak denial of service attack. All the correlated alerts are transferred to the LA for a global view and for further advanced correlation.

3.2.3 Local Analyzer (LA)

The Local Analyzer is composed of two modules (i) an Alert-Analyzing Box (ABox) and (ii) a Database Box (DBox) (figure 3.3). The **ABox** job is to receive the events and alerts from the CBox. All the CBoxes from the multiple local sites of an administrative domain send their alerts after basic correlation to the ABox. The ABox then receives these alerts and further correlates for finding strong attacks such as brute force, strong ping of death and distributed denial of service attacks. The ABox warns the grid administrator by classifying the alerts in low, medium, and high critical levels. These three types of critical levels are created by the administrator using the GUI of the GSOC. The alerts when marked with the critical levels are then saved in the DBox. The **DBox** holds information like **Security Policy** which contains all the rules created by an administrator to detect attacks for example password-cracking attempts, administrative rights gaining-attempts, log erasion, etc. **Sensor Configuration** holds all the information related to a node, for example what type of operating system is

used on a node, its kernel version number and which services are running. **Sensor Status** shows whether the node is working or not. **Local Intrusion Database (LIDB)**, a vulnerability database, holds the list of vulnerabilities from common vulnerabilities and exposures [71]. **Reported Security Alerts** are the alerts which are identified as attacks and these alerts are saved permanently in the database present in the DBox.

3.2.4 Remote Data Collector (R-CBox)

An R-CBox is a special CBox which collects events from security tools shown in figure 3.4. The R-CBoxes are working like CBoxes, the only difference is that R-CBoxes collect logs from the most important sensors in the AD. These sensors could be general security devices, proprietary softwares or customizable network monitoring systems. The purpose of collecting these logs is to double check the logs for network attacks. R-CBoxes have been introduced in the GSOC design (i) to detect attacks that target security management devices, (ii) when the attacker camouflages its attacks and (iii) when the attackers targets GSOC components. One R-CBox is deployed at every local site which takes care in case if any CBox fails. R-CBox helps to improve the security of a local site by using a variable delta (Δ) which is discussed in detail in subsection 3.5.1. The R-CBox forwards the events and alerts to the global analyzer (GA). The procedure is same as the CBox does to the LA via basic and advance correlation. The GA further investigates the received events and alerts from R-CBox to detect complex attacks at the security hosting hosts. The GA saves these alerts into the Global Intrusion Database (GIDB). The details of the GIDB are discussed in subsection 3.2.6. The GA regularly compares the events received from the CBox and R-CBox. This helps to anticipate a reaction when a critical intrusion occurs or to investigate and troubleshoot a site that could be compromised, even if a hacker erases the logs on the compromised sensors (including the security tools).

3.2.5 Global Analyzer (GA)

The Global Analyzer is the backup of the LA shown in figure 3.4. It plays its role when the LA is under an intensive distributed denial of service attack and the LA stops processing security alerts coming from CBoxes. It has both the basic and advance correlating modules. The GA also maintains a backup of running configuration of the LA. This backup is very useful to minimize the downtime due to failures of the functionalities of LA. In addition to above all, the GA receives events and alerts from R-CBoxes and compares the alerts from CBox to LA which are stored in GIDB. The GA calculate delta (Δ) to check the deviation of the normal behavior with the abnormal that is shown by the difference in the logs. A variable delta (Δ) is use to calculate this deviation which is discussed in detail in subsection 3.5.1.

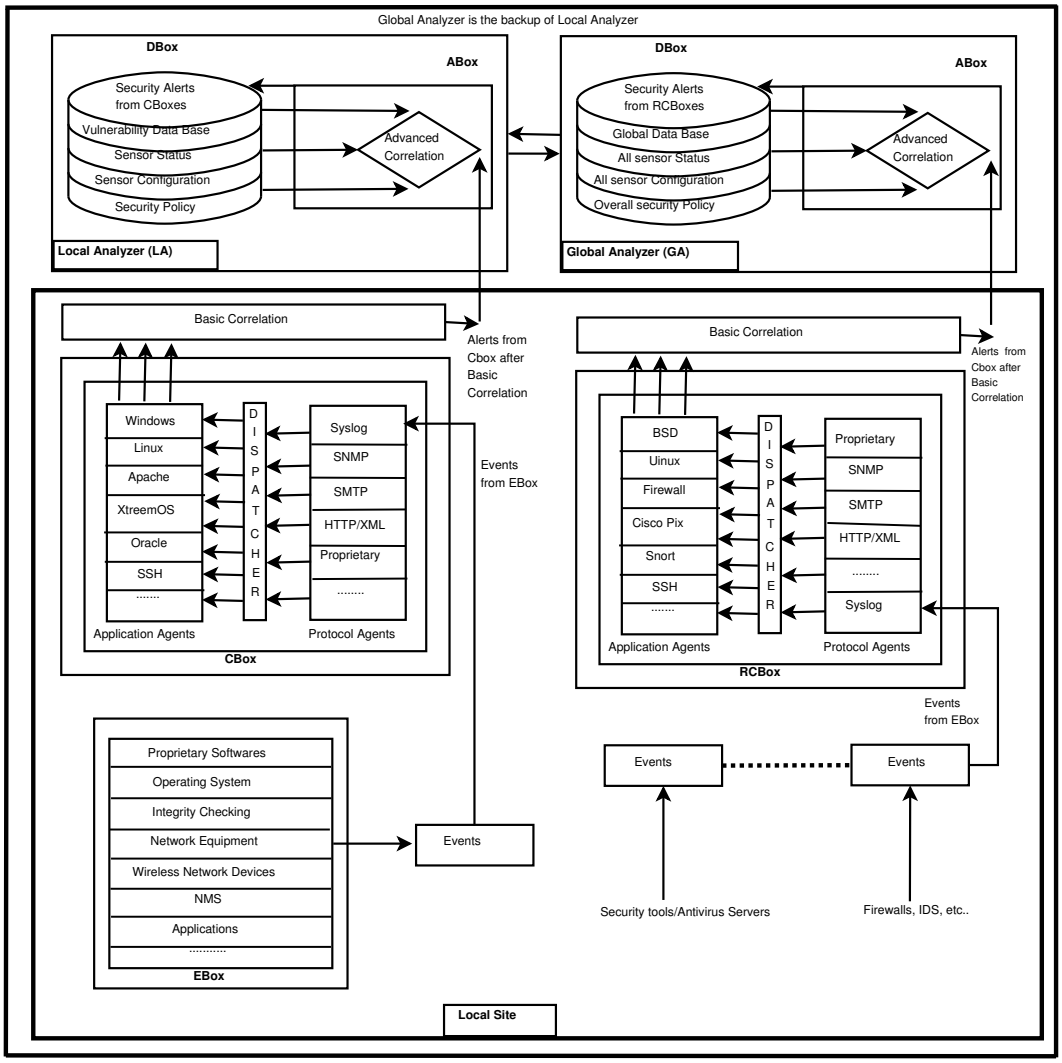


Figure 3.4: GSOC Modular Design

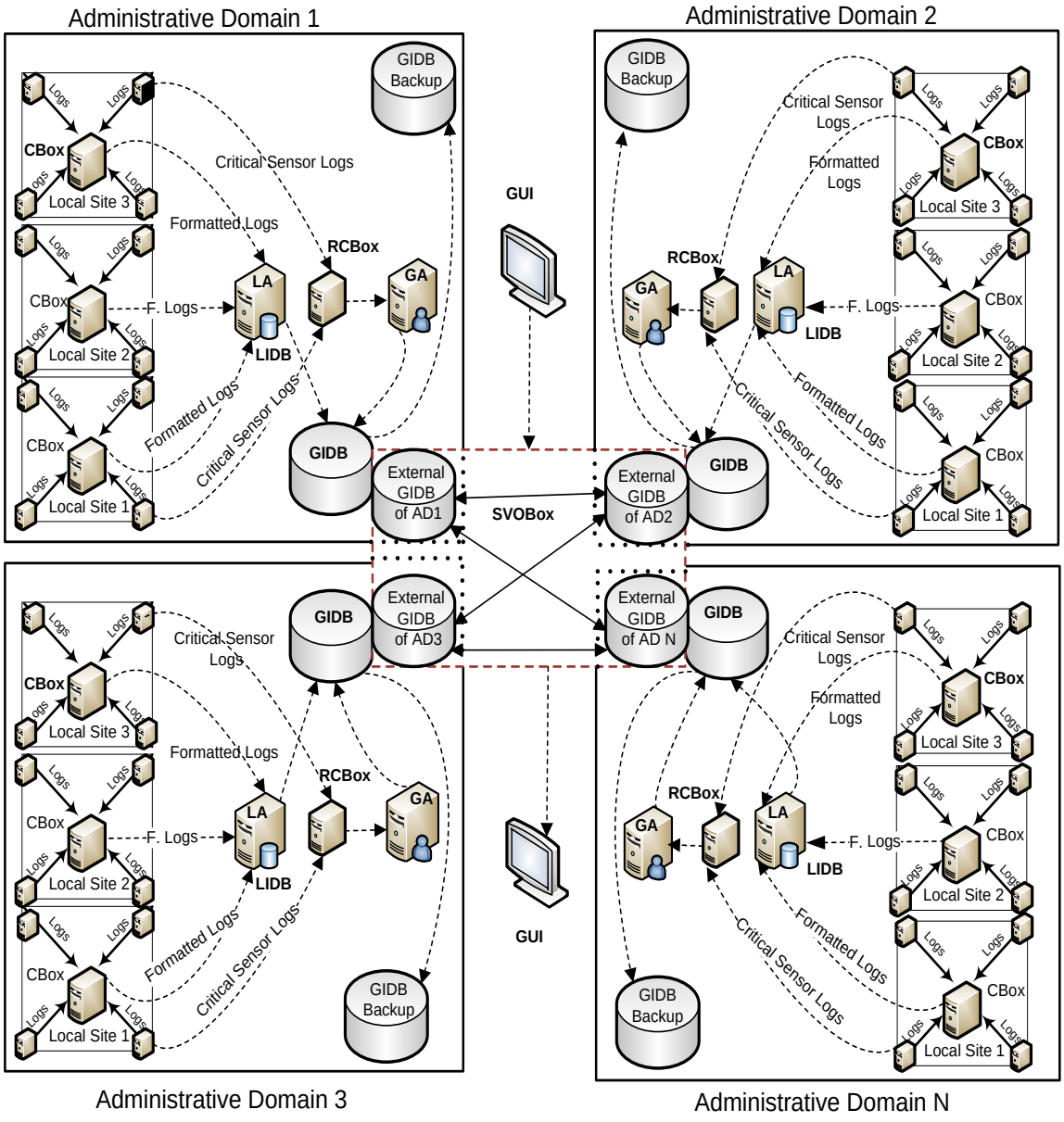


Figure 3.5: GSOC General Overview

3.2.6 Global Intrusion Detection Database (GIDB)

Figure 3.5 is a general overview of GSOC architecture and shows the main components of GSOC and their positions within the grid network. The Global Intrusion Database (GIDB) is divided in two main parts. First the internal for storing the internal information of an AD, while the second is the external which gives local security information to the other ADs of one grid computing network. The GIDB is the backup of DBox and stores all the information present in the DBox. This helps to minimize the downtime if the DBox crashes or stop functioning. In addition to provide backup, the GIDB stores the security level value which is discussed in detail in subsection 3.6. The GIDB is place where all the LAs and GA send their alerts received from CBoxes and R-CBox respectively. In the seconds part of the GIDB, the received security alerts from member ADs of the grid computing network and the local alerts which needs to be shared are kept available.

3.2.7 Secure Virtual Organization Box (SVOBox)

The Secure Virtual Organization Box (SVOBox) objective is to collect all the correlated security alerts (SA) generated in different administrative domains (AD) in a grid computing network (figure 3.6). The SVOBox assigns security level (SL) value to these ADs using a simple metric for real-time security level evaluation which has been defined by Ganame and Bourgeois [72]. Their proposed method represents criticality in three values indicated by colors (red, orange and green). Green indicates that no threat is occurring in the network. Orange indicates that threats are occurring but they are not critical at this point and red indicates intrusions are in progress which can lead to critical security problems. Their method is dedicated to multi-site network, therefore it needs some modifications before being implemented in the grid computing networks. The details of the new deployed model is discussed in section 3.6.

The main advantage of using GSOC with all its components is that it provides both the local view of the security of all the sites present in the AD and the global view of the entire grid network. GSOC is a simple software which is very easy to develop and modify-able according to any specific requirement. Additions of new features and hardwares are easily updated using application agents present in the GSOC libraries. GSOC uses the log collection mechanism which saves lots of time to set up in a network as compared to agent-based security management solutions.

3.3 Basic and Advanced Correlation

The main purpose of our correlation is to analyze complex information sequences to produce simple, synthesized, and real-time alerts. Correlation is a problem in high speed networks where data is exchanged very fast as discussed by Kruegel [73,74] and Kemmerer [75]. If the security management system is logging all the network traffic

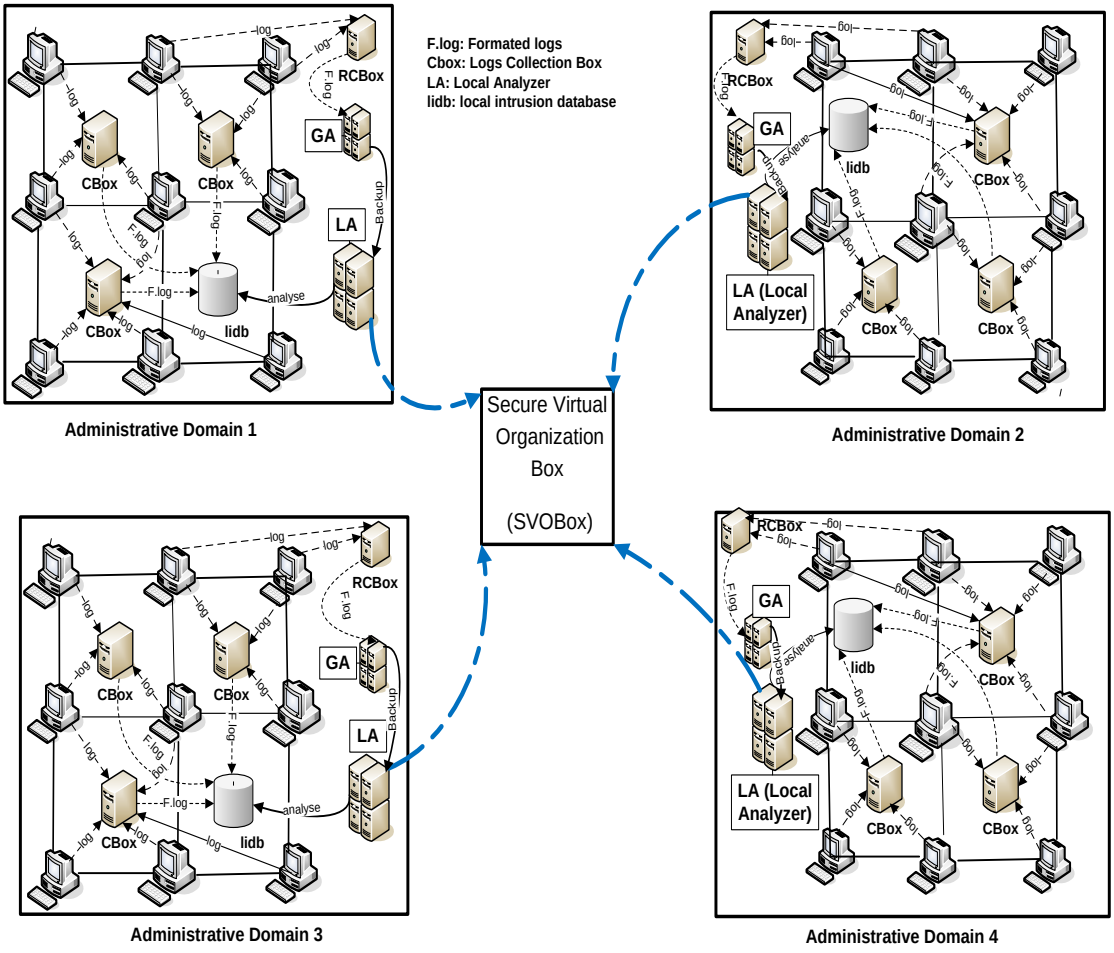


Figure 3.6: Multi-sites Network Overview

using Gigabit Ethernets then they generate enormous amount of events within the network. Precessing of these events in order to extract the security alerts, needs intense computing power and large data storages.

To achieve the objective of accurate and fast detection of alerts while handling the constraint of large flow of incoming logs, GSOC introduces a two-level correlation mechanism composed of: (i) Basic Correlation (BC) and (ii) Advanced Correlation (AC). This two-level hierarchy reduces the network traffic between GSOC components, induces an easier detection of complex intrusions and saves lots of disk space in grid computing networks. The CBox plays the role of performing basic correlation, whereas the LA is responsible for advanced correlation. Basic correlation performs few operations in order to generate attack alerts which are:

- Sequence pattern matching which identifies on-going intrusion processes, as well as complex intrusion scenarios.
- Time pattern matching which includes a new important dimension time. The CBox uses a predefined time interval (one minute) in which basic correlation is applied. It uses start time and end time to indicate the first and the last event of the same type.
- To identify duplicates and sets a specific flag in order to keep the information of duplicates. These duplicate event posses exactly the same fields from one event to another except for the time field.
- Threshold comparison including comparison of the collected alerts with a premeasured threshold.

The decision on whether an on-going attack is happening is made by this BC engine using the above-mentioned criteria. An accumulated number is generated, which reduces the number of alerts and represents the number of times the source attacked the target. This field can be understood as a number of duplicated alerts. The main purpose of BC is to reduce the network load between the GSOC modules; therefore attack detection is easier to perform because there are two-level hierarchy analyzers. BC does not have the ability to detect distributed denial of service or strong brute force attacks. The CBox is capable of detecting only weak attacks. For example, if two attackers are simultaneously attacking one target sensor in an AD, performing a DDoS or strong brute force attack, the CBox will report one alarm for a DoS attack and one alarm for a weak brute force attack, originating from two different attackers. The task of deciding whether it is a DDoS attack or any other kind of strong attack is dedicated to the LA, more specifically to the ABox.

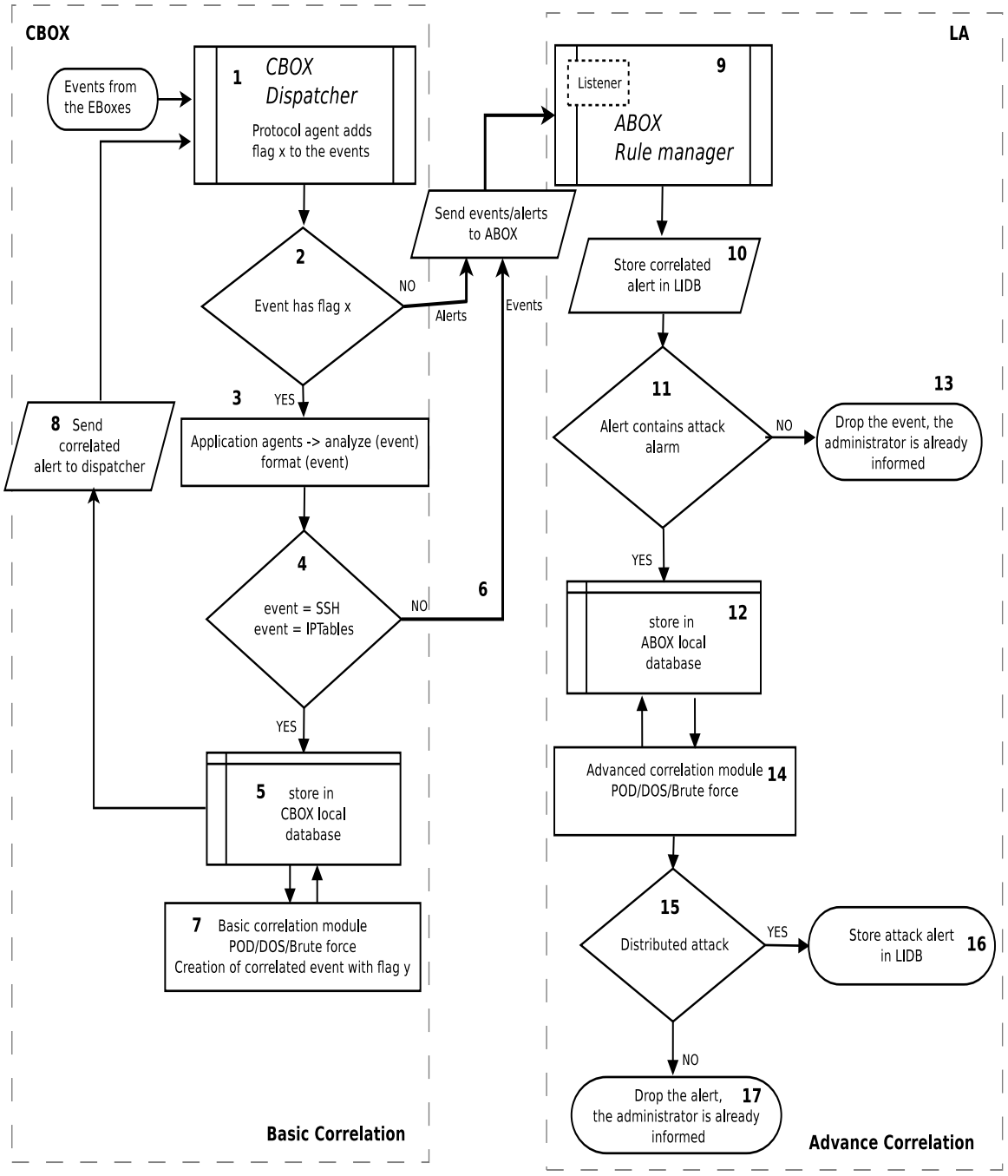


Figure 3.7: Basic and Advanced Correlation Flow Chart

The left part of figure 3.7 is the detailed explanation of basic correlation. The basic correlation module can be considered as an event marker. Each event is labeled depending on its contents, whether it is containing an attack alert or not.

1- Each event sent from an EBox is received by the protocol agents at the CBox, and then this event is labeled with a flag (eg: flag x).

2- This flag points out to the dispatcher that this event should be analyzed first by the application agents.

3- If the received event matches with the security rule, it will be standardized by internal formatting.

4- The dispatcher inspects whether these formatted events are the ones that the administrator is interested in correlating (event originating from ssh session or event from IPtable rules at the EBoxes).

5- If it is the case, then these kinds of events are stored in a local database for a very short period of time (at most one minute). The program deletes old events with a simple SQL query.

6- If it is not, then the CBox forwards the events to the LA (specifically the ABox) in order to display the reported alert as a piece of information to the administrator. Even though these events are not correlated, they have passed through the application agent's analysis, this means that the administrator might be interested in knowing the minor activities. These kinds of event can also be correlated in future to accomplish network traffic reduction.

7- When the correlation finishes the events are now considered as the alerts. The stored alerts in the local database are marked with the new flag (eg: flag y). This flag is different than the one added to the event (refer to step 2).

8- This flag tells the dispatcher that this alert has already correlated and should be sent to the LA for further analysis. Afterwards, only the correlated alerts are stored in the local database and transferred to the dispatcher which further forwards them to the LA.

At this stage, alerts that contain an attack are forwarded to the LA including those that are attack-free. The communication between the CBox and the LA is over socket protocol. A simplified view of the fields of the correlated alert structure which exchanges the alert between the CBox and the LA is described in figure 3.8. The tables which are involved in the creation of the alerts are, **Message Table :** includes all the details of the events received at the CBox. **Message Type Table :** contains human-readable description of Message Type ID. **Host Table :** identifies each host of the grid computing network that the security system monitors and checks its state (running or down). **Host Types :** contains a human-readable description of each host type.

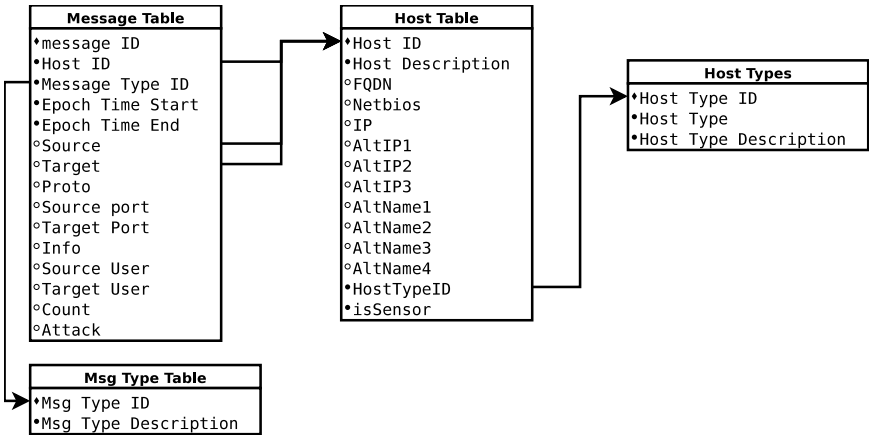


Figure 3.8: Simplified View of the Composition of the Formatted Alert

The right side of figure 3.7 explains the advanced correlation of the LA, performed in the ABox.

9- The CBox sends events and alerts to the ABox, and the listener module in the ABox accepts the correlated alerts from the CBox. When a correlated alert arrives at the ABox, a rule manager first checks if the network administrator is interested in monitoring the information about the sensor (monitored device) included in the alert.

10- If it is the case, the alert is stored in the LIDB and reported to the administrator. However, at this stage the administrator still does not have a clear view of the strong attacks on any of its sensors.

11- For this reason, it checks if the alerts that contain an alarm of the attacks (BF, DoS & DDoS) need to be correlated further.

12- If it is the case, then they are stored in a local database for a short period of time (at most one minute, just like the local database at the CBox (refer to step 5)) until the advanced correlation finishes.

13- If it is not, then the events without alarms are dropped, because the administrator has already been informed.

14- The operations for performing the advanced correlation task are (i) target and (ii) time correlation. This module counts the number of notifications for the same target from different sources (attackers) within the time interval (equal to one minute).

15- If there is more than one attacker assaulting the same target (sensor) in the same unit of time (the threshold value is one minute). It generates the strong attack alert.

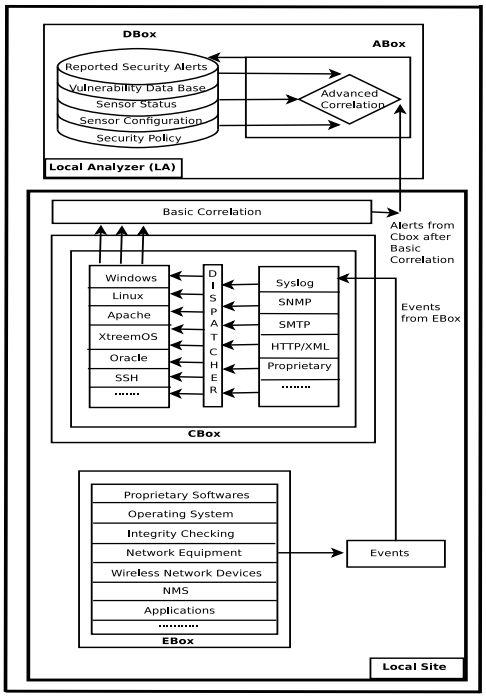


Figure 3.9: Inter Communication view of EBox, CBox and Local Analyzer

16- One alert of a strong attack alert is generated and stored in the LIDB, which is also displayed at the GUI of the GSOC.

17- If not, then the alert is dropped as the administrator has already been informed about this event.

3.4 Security Alert Generating Mechanism

EBoxes (see figure 3.9) are the source of data for GSOC and cover a wide range of devices, from a normal computer to any device in the network. The nice feature is that no additional software is required to be installed on the EBoxes in order to send logs. To integrate a new EBox in GSOC, a simple configuration has to be made, such as a log redirection. One CBox collects data from multiple EBoxes using protocol agents. The logs are processed depending on their protocol and forwarded to the dispatcher which sends them to the appropriate application agent. The application agents extract the information from the logs and create formatted events which are sent to the basic correlation module. One application agent is required for each type of log collected. The Basic Correlation (BC) module has a defined time limit which holds the reported security alerts for one minute and correlates for detecting attacks. The Basic Correlation module forwards the correlated attack information to the Advanced Corre-

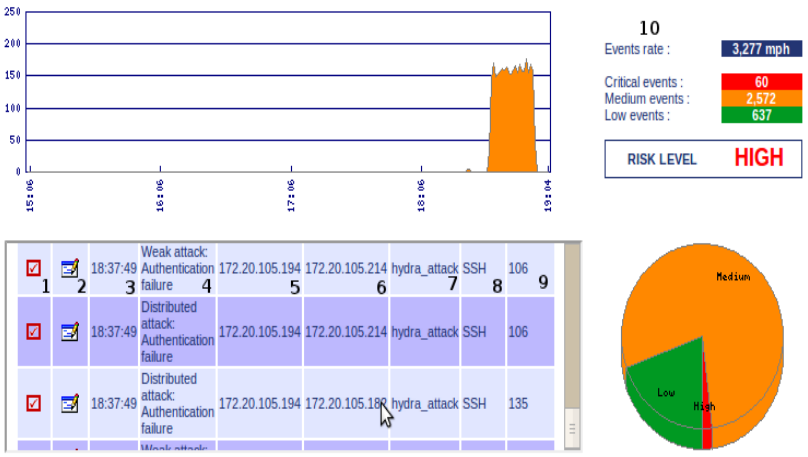


Figure 3.10: Alert Reported at the Main Dashboard of the GSOC

lation (AC) module for detecting distributed attacks. The advanced correlation module collects logs from multiple CBoxes and further correlates for more sophisticated and distributed attacks. If an attack is detected, the ABox reports it to the administrator in real time and saves the security alert permanently in the DBox. The same reported alert will also be saved in the Global Intrusion Database (GIDB) as a backup of DBox (see figure 3.12).

Few examples of the reported alerts at the GSOC's dashboard are shown in the figure 3.10. (1) Starting from the left column which shows the criticality of the alert in red color. (2) The next column displays an icon, upon selection can show the details of the attacks. (3) The next column shows the time at which the attack was occurred. (4) The next column shows the type of attack, in the figure it shows the different types of reported attacks. (5&6) The next two columns show the IP address of the attacker and the victim machines. (7) The next column shows the name of the user by which the attacker tries to do the attack, in the figure it is hydra_attack. (8) The next column shows the name of the protocol used for launching the attack. (9) In the last column the number of attempts made by the attacker in one minute time window. (10) The history of the alerts are displayed which occurred during last one hour according to their criticality.

3.5 GSOC Internal Architectural View

In figure 3.12 the core architectural view of GSOC and the internal design of each component is shown. It shows all the seven components with their interconnection. (i) The GSOC starts by processing the logs sent from EBox using protocol agents

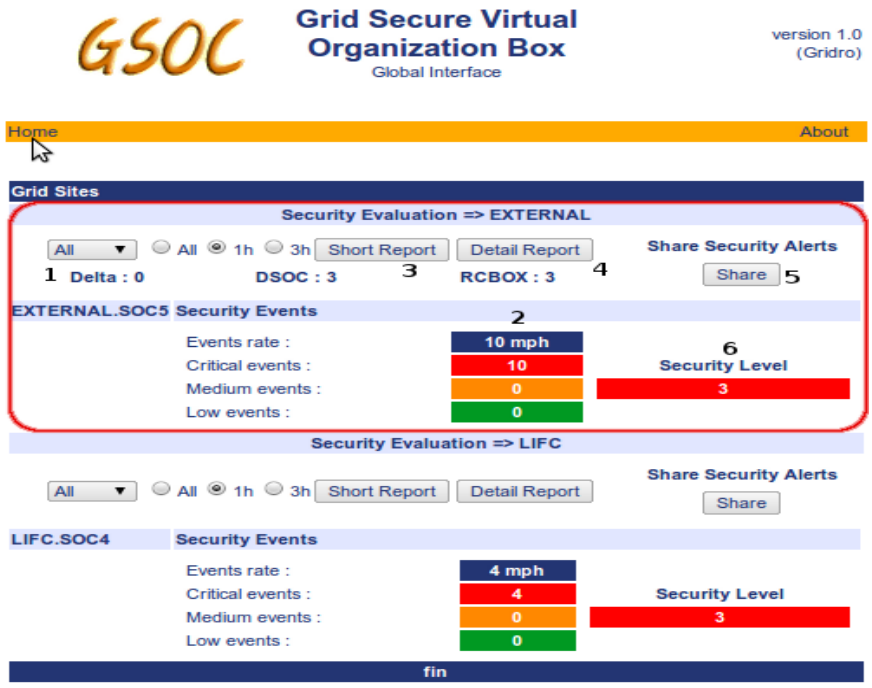


Figure 3.11: SVOBox Main Dashboard

to the CBox. (ii) The CBox receives the logs via dispatcher and sends them to the application agents to detect the specific events. The formated events are forwarded to the basic correlation module for detecting the attack incidents. After one minute they are forwarded to the advance correlation module present in the LA. (iii) The LA further correlates the received alerts and make sure that it has detected the real attack. It saves the reported alerts in the DBox and sends a copy to the GIDB. (iv) The R-CBox forwards the most important EBoxes logs to the GA using basic and advance correlation modules. (v) The GA forwards the received alert to the to GIDB and saved them permanently. (vi) The GIDB stores all the alerts received from local sites and calculates the delta (Δ). GIDB also provides a mechanism to share the security alerts with other member of the grid computing network using external GDIB. (vii) The SVOBox is used for security evaluation of each member of the grid computing network, using the security alerts information provided by them.

3.5.1 Calculating Delta (Δ)

Figure 3.11 shows the (1) delta (Δ), the messages received by the R-CBoxes, (2) the messages received from local sites, (3) a short report, (4) a detailed report, (5) the security alert sharing option and (6) a security evaluation of two ADs. At the final stage the SVOBox evaluates the security level value for each AD. Security evaluation

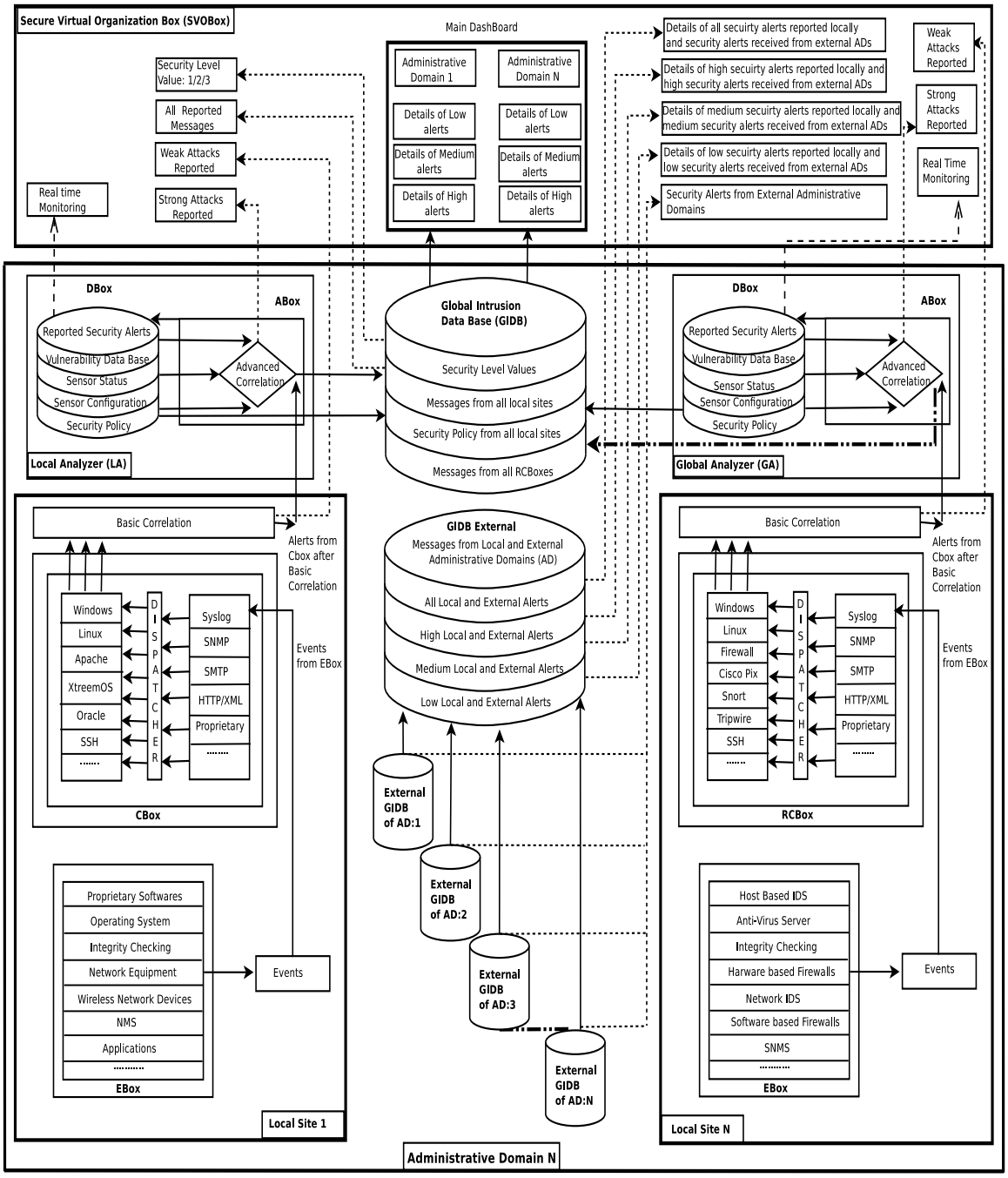


Figure 3.12: GSOC Internal Architectural View

is discussed in detail in section 3.6.

The R-CBox provides a way to calculate the delta (Δ) by the logs generated during normal operation of the network with the abnormal behavior using GIDB. The delta is calculated by comparing the events received from all local sites and the events received from all R-CBoxes in the GIDB as shown in figure 3.12. For one AD the delta (Δ) is calculated using the formulas shown below:

$$Number\ of\ alerts(A)\ received\ at\ CBox\ at\ time(t) = \sum_{A=1}^{\infty} CBox_A(t) \tag{3.1}$$

$$Number\ of\ alerts(A)\ received\ at\ RCBox\ at\ time(t) = \sum_{A=1}^{\infty} RCBox_A(t) \tag{3.2}$$

$$Delta => \sqrt{(\Delta)^2} = \sum_{A=1}^{\infty} CBox_A(t) - \sum_{A=1}^{\infty} RCBox_A(t) \tag{3.3}$$

$$Possibilities\ of\ attacks\ are \propto to\ the\ value\ of\ \Delta \tag{3.4}$$

The greater the value of delta the greater are the chances that the malicious activities are in progress. The delta is directly proportional to the possibility of attacks. The delta (Δ) shows that either the local site is under attack or any CBox has stopped processing the alerts. This is possible when the attackers delete the logs coming from different sensors or the attackers target any CBox. The same applies when the attackers target and R-CBox. This mechanism provides a fault tolerance capability in the GSOC design which helps to detect more sophisticated attacks.

3.5.2 Security Alert Sharing

The lower part of GIDB is called External GIDB and it is responsible for sharing the reported security alerts with other ADs in a grid computing network. Security alert sharing can be very effective in blocking cross-domain attacks while keeping in view the composition of grid computing network. As discussed earlier in grid computing network different ADs each of consist of multi-local sites are combined together to form a network. Security alert sharing feature informs the participating member of a grid network about the nature of attacks that are under progress in any ADs. One solution to this problem is to provide a way to share, between different ADs, their security

alerts. But, sharing all security alerts of one entire network with another administrative is something that security policy should forbid. To address this issue GSOC provides a mechanism by which the network administrator of an AD can share some security alerts with selected ADs depending on their security level. Security alerts belong to three categories low, medium and high which rank the confidentiality of the alert. The details of each category is discussed in detail in section 3.6. Figure 3.13 shows the overview of the reported security alert sharing mechanism between six administrative domains.

SVOBox assigns security level values by manipulating the number of security alerts generated within the local sites of the AD. All the security alerts generated in different ADs of a grid network can be seen at the SVOBox main dashboard. The SVOBox dashboard access is granted to the network administrator of each AD and only the ADs which are sharing their resources with each other are allowed to send and receive the security alerts between them (figure 3.11).

3.6 Security Evaluation of Administrative Domains

Security evaluation of administrative domains that form a grid computing network is required in order to create trust between ADs. Security levels are a representation of the current security state of the AD. Evaluating the security of an AD which can be a multi-sites network is a complex task as it includes many heterogeneous devices. Security can be evaluated using multiple metrics. Some possibilities are:

(i) The skills of the IT security team that includes experience, certifications, trainings, and their availability.

(ii) The security devices and softwares used including (i.e., firewalls, intrusion detection and prevention systems (IDPS) and anti-virus softwares). Their configuration level includes the security rules and policies that must be updated on time.

(iii) Security contingency plans in case of severe attacks.

(iv) Time required to restore all the systems and services after shutdown in case of disasters.

(v) User awareness programs, for maintaining their system's security.

(vi) Reports of deep vulnerability scans, security scans, and pen tests must also be considered.

All the above mentioned aspects needs to be evaluated by a third party for non-biased evaluation results. A detailed view of the security evaluation has been presented in [72]. For evaluating security level values in GSOC, some equations written below

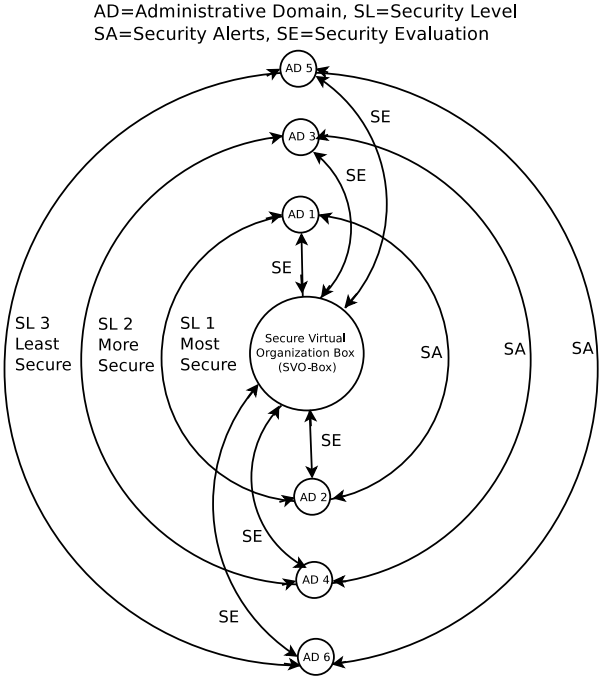


Figure 3.13: GSOC Security Alert Sharing Mechanism

have been applied to any AD which is a part of the grid computing network. These equations help GSOC to categorize the ADs by processing the number of low, medium and high alerts generated within their local sites. These levels are criticality levels that are assigned by to each alert by the network administrator of each AD. GSOC categorizes the ADs according to three security levels. Security level one is the highest security level while three is the lowest one (see figure 3.13).

3.6.1 Static and Dynamic Security Evaluation

In GSOC the evaluation is done in two phases simultaneously, first is the static evaluation done by the tools Nessus [76], OpenVAS [77] and Saint [78]. These tools are installed on the network and executed to detect the vulnerabilities present in the machines. Most of the security evaluation tools uses few steps as explained from 1 to 3 to detect the vulnerabilities in the network. (1) They search the machines which are running TCP & UDP services on the network. (2) For each service they uses a set of probes to detect some vulnerabilities present in the system. These probes could be small attacks of DoS, users rights gaining attempts or collecting some sensitive information from network devices. (3) The detected vulnerabilities are then categorized in different levels and displayed at the dashboards of the tools. In the experiments shown in the table 3.1 the vendors of each tool have detected few vulnerabilities and catego-

Table 3.1: Security Evaluation Comparison of 10 Machines

	OpenVas			Nessus			Saint		
M #	**LA**	**MA**	**HA**	**LA**	**MA**	**HA**	**LA**	**MA**	**HA**
M1	10	1	1	27	4	0	8	2	3
M2	10	1	1	27	4	0	11	2	0
M3	10	1	1	27	4	0	7	2	2
M4	28	4	1	96	16	0	14	2	3
M5	10	1	0	21	3	0	7	3	3
M6	10	1	1	26	4	0	8	2	1
M7	10	1	1	27	4	0	8	2	1
M8	10	1	1	26	4	0	8	2	0
M9	10	1	1	27	4	0	8	2	2
M10	7	1	0	30	2	0	7	1	1
Total	115	13	8	334	49	0	86	20	16

LA=Low Alerts, MA=Medium Alerts, HA=High Alerts, M#=Machine Number

rized them as low, medium and high. Table 3.1 shows the number of security alerts detected during scan on 10 machines. The configuration of the machines are CPU 2.4 GHz and RAM 512 MB which were connected to the network in the lab as shown in figure 4.7.

The second is the dynamic evaluation which is done by the GSOC using the value of the delta as explained in section 3.5.1. Therefore for security evaluation in GSOC the results of these three tools are also incorporated with the dynamic security evaluation to give a better view of the security of the entire network.

3.6.2 Security Cases under Different Attack Scenarios

LAV (Low Alert Value) is the number of low level alerts reported which are, for example, information like session opening, session closing, or services start/stop/restart. MAV (Medium Alert Value) is the number of medium alerts reported, and HAV (High Alert Value) is the number of high alerts reported in any AD. The behavior of the AD can be categorized according to three case which are observed during eight hours of tests as explained below:

Case-I: When an AD operates under normal circumstances, LAV is always greater than MAV which is greater than HAV ($LAV > MAV > HAV$).

Case-II: There is a slight change in the normal behavior of an AD or in any one of its local sites. This happens when the inexperienced attackers try to launch basic attacks. Some examples might be, manual use of incorrect password attempts, port scans, and ICMP flooding. They use their personal machines without IP spoofing for launching these attacks, making these attacks easily detectable.

Case-III: There is a major change in the normal behavior of an AD or in any one of its local sites. This case occurs when the experienced attackers use multiple machines for launching distributed attacks. They use automated tools and spoof their IPs. The duration of the attacks last for a very long period of time and they mix their attacks with normal behavior to camouflage their operations. These attacks are difficult to detect and very destructive in nature. Some of the examples of these attacks are the use of brute force attacks using automated tools such as THC Hydra [59] & Guess Who [60], DoS & DDoS attacks using Slowloris [79], and UDP & TCP flooding by changing the maximum transmission unit (MTU) of the packets using hping [80].

All the three case are applied in our lab and in the Grid'5000 network. The obtained results are discussed in detail in Chapter 4.

NOTE: The standard policy must be adopted by all members of the grid computing network for assigning the critical values to the attacks in order to have a homogeneous reporting systems.

3.6.3 Formalization of Dynamic Security Evaluation & Security Level Assignment to the AD

SL = Security Level. **GN** = Grid Network. **LS** = Local Site.
LA = Low Alerts. **MA** = Medium Alerts. **HA** = High Alerts.
LAV = Low Alerts Value. **MAV** = Medium Alert Value.
HAV = High Alert Value. **LSA** = Local Sites Alerts.
ll = low level, **ml** = medium level and **hl** = high level.
t = time at which the alerts detected.
i = identification of the LS within AD.
j = total number of LS.

Let GN be a grid computing network:

$$GN = \{AD_1, AD_2, ...AD_m\}$$

Algorithm 1 : Security Level Assignment to the Administrative Domains (ADs)

for all $(AD\ in\ GN)$ **do**
  if $(LAV_{AD_i} < MAV_{AD_i}\ and\ LAV_{AD_i} > HAV_{AD_i})$ **then**
    SL = 2
  else if $(LAV_{AD_i} > MAV_{AD_i}\ and\ LAV_{AD_i} < HAV_{AD_i})$ **then**
    SL = 3
  else if $(LAV_{AD_i} > MAV_{AD_i}\ and\ MAV_{AD_i} > HAV_{AD_i})$ **then**
    SL = 1
  else if $(MAV_{AD_i} < LAV_{AD_i}\ and\ HAV_{AD_i} < LAV_{AD_i})$ **then**
    SL = 2
  else if $(MAV_{AD_i} > LAV_{AD_i}\ and\ MAV_{AD_i} < HAV_{AD_i})$ **then**
    SL = 3
  else$(MAV_{AD_i} > LAV_{AD_i}\ and\ MAV_{AD_i} > HAV_{AD_i})$
    SL = 3
  end if
end for

with

$$AD_i = \{LS_i^1, LS_i^2, ...LS_i^{n_i}\}$$

where n_i is the number of local sites of the administrative domain i, and m is the number of administrative domains in GN. Now LAV, MAV and HAV at time "t" can be determined by:

$$LAV\ is\ give\ by\ LA_{AD_i}(t) = \sum_{j=1}^{n_i} LSA_{LS_i^j}(ll, t)$$

where $LSA_{LS_i^j}(ll, t)$ is the low level value at time "t" of the total local sites "j" at "AD_i". LAV is the number of total low alerts reported within all local sites of an AD.

$$MAV\ is\ give\ by\ MA_{ADi}(t) = \sum_{j=1}^{n_i} LSA_{LS_i^j}(ml, t)$$

where $LSA_{LS_i^j}(ml, t)$ is the low level value at time "t" of the total local sites "j" at "AD_i". MAV is the number of total medium alerts reported within all local sites of an AD.

$$HAV\ is\ give\ by\ HA_{ADi}(t) = \sum_{j=1}^{n_i} LSA_{LS_i^j}(hl, t)$$

where $LSA_{LS_i^j}(hl, t)$ is the low level value at time "t" of the total local sites "j" at

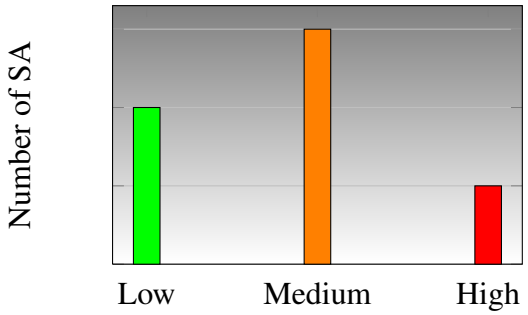


Figure 3.14: Assignment of Security Level 2

"AD_i". HAV is the number of total high alerts reported within all local sites of an AD.

Now that LAV, MAV and HAV have been defined, each AD security level can be set according to the equations 3.5 to 3.10 .

The equation 3.5 shows that the AD has medium security alerts greater than low and high, whereas low security alerts are greater than high security alerts, therefore it should be placed in SL2 (Figure 3.14).

$$\boxed{if(LAV_{AD_i} < MAV_{AD_i}\ and\ LAV_{AD_i} > HAV_{AD_i})\ then\ SL = 2} \tag{3.5}$$

Equation 3.6 shows that the AD has high security alerts greater than low and medium, whereas low security alerts are greater than medium security alerts; therefore it should be placed in SL3 (Figure 3.15).

$$\boxed{elseif(LAV_{AD_i} > MAV_{AD_i}\ and\ LAV_{AD_i} < HAV_{AD_i})\ then\ SL = 3} \tag{3.6}$$

Equation 3.7 shows that the AD has low security alerts greater than high and medium, whereas medium security alerts are greater than high security alerts; therefore it should be placed in SL1 (Figure 3.16).

$$\boxed{elseif(LAV_{AD_i} > MAV_{AD_i}\ and\ MAV_{AD_i} > HAV_{AD_i})\ then\ SL = 1} \tag{3.7}$$

Equation 3.8 shows that the AD has low security alerts greater than high and medium, whereas high security alerts are greater than medium security alerts; therefore

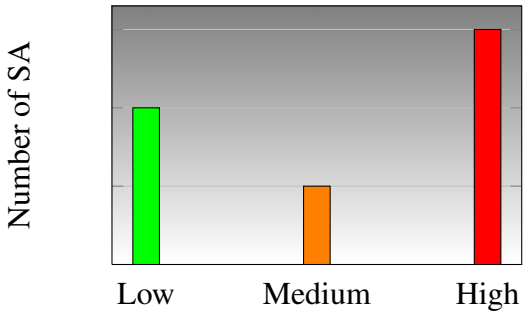


Figure 3.15: Assignment of Security Level 3

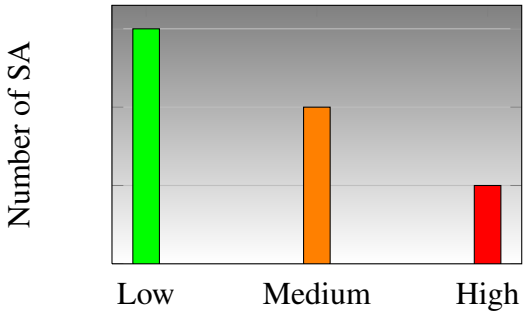


Figure 3.16: Assignment of Security Level 1

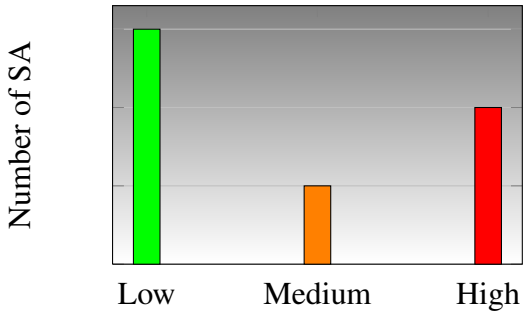


Figure 3.17: Assignment of Security Level 2

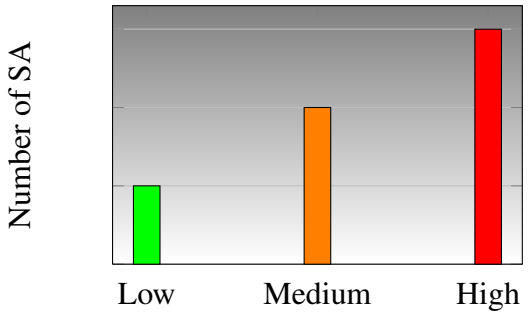


Figure 3.18: Assignment of Security Level 3

it should be placed in SL2 (Figure 3.17).

$$\boxed{elseif(MAV_{AD_i} < LAV_{AD_i} \; and \; HAV_{AD_i} < LAV_{AD_i}) \; then \; SL = 2} \tag{3.8}$$

Equation 3.9 shows that the AD has high security alerts greater than low and medium, whereas medium security alerts are greater than low security alerts; therefore it should be placed in SL3 (Figure 3.18).

$$\boxed{elseif(MAV_{AD_i} > LAV_{AD_i} \; and \; MAV_{AD_i} < HAV_{AD_i}) \; then \; SL = 3} \tag{3.9}$$

Equation 3.10 shows that the AD has medium security alerts greater than low and high, whereas high security alerts are greater than low security alerts; therefore it

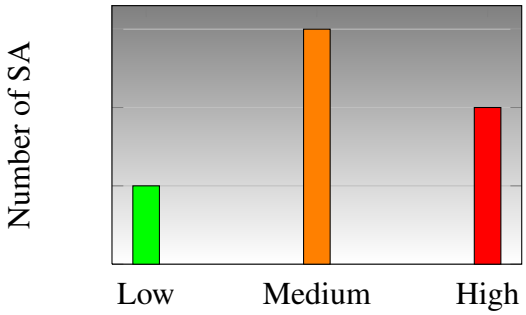


Figure 3.19: Assignment of Security Level 3

should be placed in SL3 (Figure 3.19).

$$\boxed{elseif(MAV_{AD_i} > LAV_{AD_i}\ and\ MAV_{AD_i} > HAV_{AD_i})\ then\ SL = 3} \quad (3.10)$$

Figure 3.20 is the graphical representation of the scan performed by Nessus, OpenVAS and Saint to obtained the static evaluation of the security on 10 machines. These scans are taken during the normal operation of the network in the lab as shown in figure 4.7. The results show that the low alerts are greater than medium and medium are greater than high alerts. The objective for the static evaluation is to make sure that the results obtained must verifies the equations that are proposed for the dynamic security evaluation in the GSOC. This static security evaluation validates that the equations that are used for dynamic security evaluation in GSOC complying each others results.

3.7 Introduction to XtreemOS

The XtreemOS is an open source software kit [81][1]. It is developed as an alternative to the grid middlewares and it provides single sign on feature to its users. The users needs to login only at once and after getting success they are allowed to use any resources according to their assigned rights [10]. The idea of the development of XtreemOS was to make the use of open source operating systems by introducing some changes in the kernel. These changes has been embedded in XtreemOS in the form of grid services. The tasks which are done using the grid middlewares can now be done by using the services added in the XtreemOS [82]. This makes the job of the grid users easier because using XtreemOS, the users need to do the configurations only in one operating system throughout the grid network. The XtreemOS provides three services to its users namely: application execution management, data management, and vir-

[1] http://xtreemos.org/

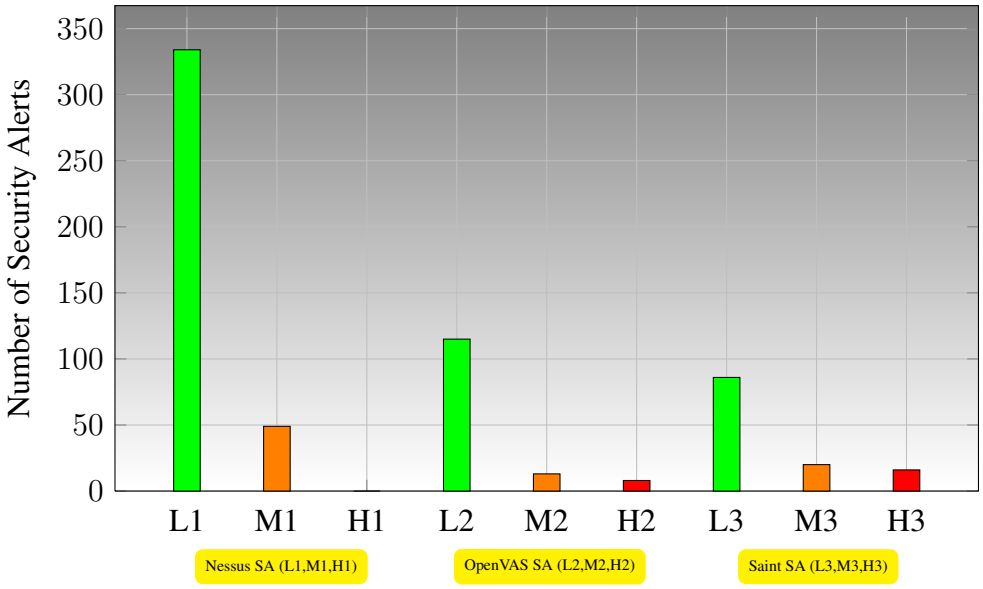


Figure 3.20: Security Alert Statistics of 10 Machines using Nessus, OpenVAS and Saint

tual organization management. It enables XtreemOS to run three types of devices that are PCs (Linux-XOS), clusters and PDAs (XtreemOS-MD). XtreemOS for PCs support all kinds of fixed and laptops machines. The XtreemOS cluster type is based on Linux Single System Image (LinuxSSI) which takes the advantage of Kerrighed SSI technology [83]. It represents a huge set of computing sources as a single entity [10]. The XtreemOS-MD supports mobile device to connect to the Virtual Organizations (VO) [10].

XtreemOS is meant to develop to support Virtual Organization (VO) in grid computing networks therefore it provides full support to Virtual Organizations (VOs) [84]. A VO is a group of ADs that are formed to achieve a specific objective for a short period of time or permanently. When the VO is initiated, the VO manager sometimes creates new policies, which are further distributed to the other members of the VO. These policies may include changes in the security settings in their respective networks such as allowing some applications to make the connections with the remote machines or services. These settings creates security holes in the network which can be exploited by the attackers [85].

3.7.1 XtreemOS Architecture & Services

XtreemOS has two main layers, the XtreemOS Foundation layer (XtreemOS-F) and the XtreemOS Grid Support layer (XtreemOS-G) as shown in figure 3.21. The XtreemOS-F layer supports PCs for single sign on, cluster based on Kerrighed sys-

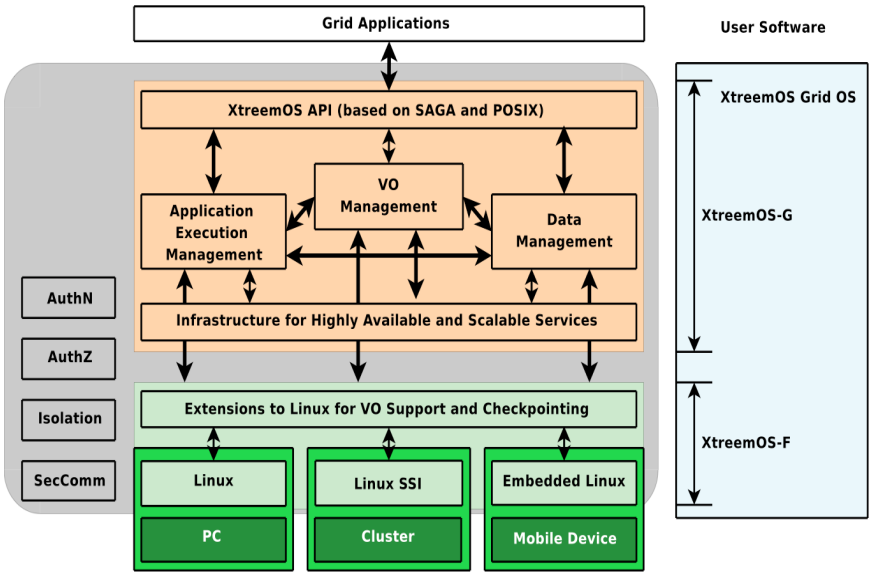


Figure 3.21: XtreemOS Architecture [10]

tems and the mobile devices. The XtreemOS-G supports service such as Application Execution Management (AEM), data management using grid file system (XtreemFS), and application and VO managemnt & security [82].

XtreemOS has three kinds of node configurations namely, (i) Core, (ii) Resource and (iii) Client. (i) The Core nodes are responsible for providing the services to the other nodes so that they can provide the resources to the XtreemOS systems. (ii) The Resource nodes are responsible for providing the resources to the XtreemOS systems. (iii) The Client nodes uses the resources provided by XtreemOS.

3.7.2 XtreemOS Security Issues

XtreemOS is based on Linux, where access to kernel needs authentication and authorization. In XtreemOS the features of the middlewares are embedded in the kernel of the operating system. The local and remote users of XtreemOS can have access to many resources of the systems using single sign on. It includes access to the kernel of the systems. This weakens the security of the individual systems and makes the network more vulnerable to attacks. From the security point of view, the services which run on the core node are very important. Stopping any service without proper approval can lead to serious breach of security which could have unrecoverable consequences [86]. There exist many services on the core node. In this thesis only the Metadata Replica Catalog (MRC) and Directory Service (DIR) are covered [85].

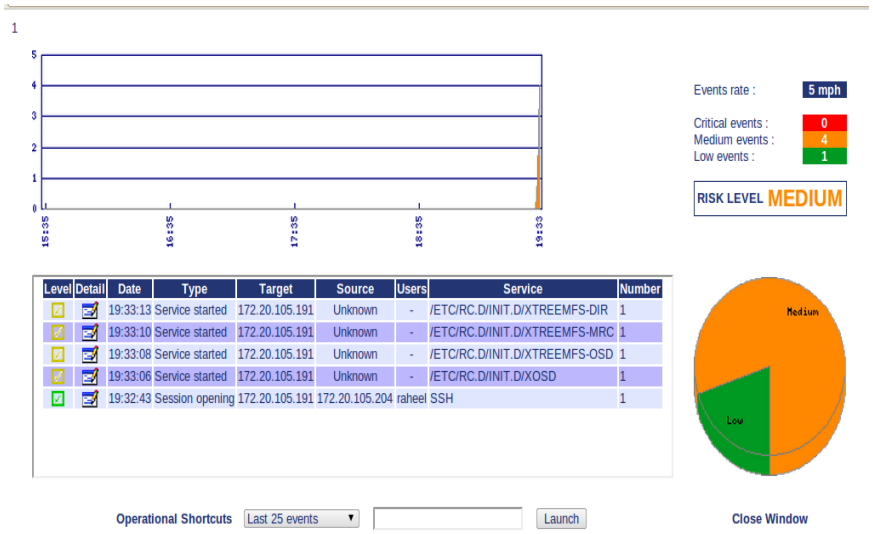


Figure 3.22: XtreemOS Dashboard

3.7.2.1 Metadata Replica Catalog (MRC)

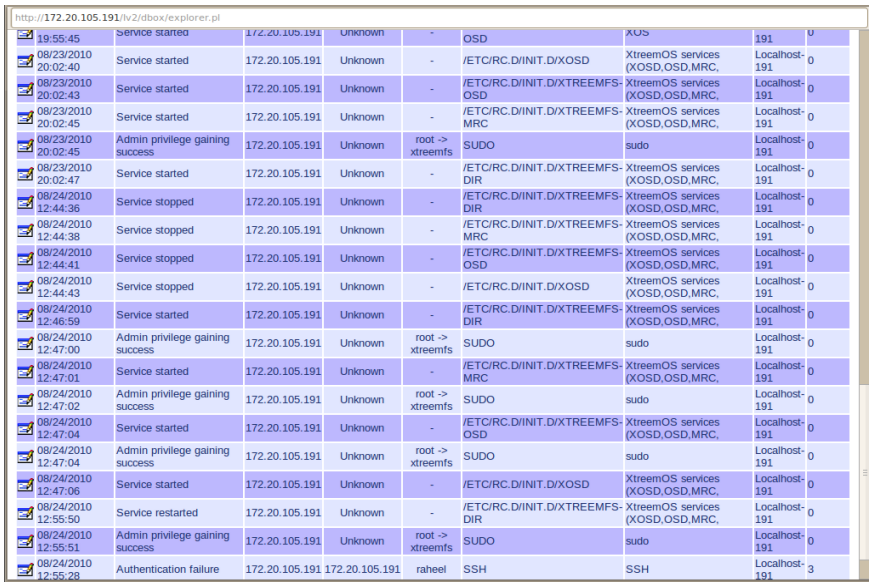
The MRC service authenticates and authorizes users to have access to the files. It contains metadata of the directory tree such as file names with their size and modification time [85].

3.7.2.2 Directory Service (DIR)

The DIR service is the main service which has all the information of the other services running in the XtreemFS. It helps the MRC service to identify the storage servers [85].

3.7.3 Monitoring the Security of the XtreemOS using GSOC

In our experiments shown in figures 3.22 and 3.23, GSOC detects if any of the services discussed above stopped working by DDoS attacks. Figure 3.22 is the dashboard that shows the status of the attacks at real time. It shows the date and time when and which service is started. Figure 3.23 is the detailed view of each reported attack. It shows more details of each alert and more history of the alerts when they were reported.



http://172.20.105.191/lv2/dbox/explorer.pl

19:55:45	Service started	172.20.105.191	Unknown	-	OSD	XOS	191	0
08/23/2010 20:02:40	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XOSD	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/23/2010 20:02:43	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-OSD	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/23/2010 20:02:45	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-MRC	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/23/2010 20:02:45	Admin privilege gaining success	172.20.105.191	Unknown	root -> xtreemfs	SUDO	sudo	Localhost-191	0
08/23/2010 20:02:47	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-DIR	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:44:36	Service stopped	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-DIR	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:44:38	Service stopped	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-MRC	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:44:41	Service stopped	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-OSD	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:44:43	Service stopped	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XOSD	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:46:59	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-DIR	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:47:00	Admin privilege gaining success	172.20.105.191	Unknown	root -> xtreemfs	SUDO	sudo	Localhost-191	0
08/24/2010 12:47:01	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-MRC	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:47:02	Admin privilege gaining success	172.20.105.191	Unknown	root -> xtreemfs	SUDO	sudo	Localhost-191	0
08/24/2010 12:47:04	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-OSD	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:47:04	Admin privilege gaining success	172.20.105.191	Unknown	root -> xtreemfs	SUDO	sudo	Localhost-191	0
08/24/2010 12:47:06	Service started	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XOSD	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:55:50	Service restarted	172.20.105.191	Unknown	-	/ETC/RC.D/INIT.D/XTREEMFS-DIR	XtreemOS services (XOSD,OSD,MRC,	Localhost-191	0
08/24/2010 12:55:51	Admin privilege gaining success	172.20.105.191	Unknown	root -> xtreemfs	SUDO	sudo	Localhost-191	0
08/24/2010 12:55:28	Authentication failure	172.20.105.191	172.20.105.191	raheel	SSH	SSH	Localhost-191	3

Figure 3.23: XtreemOS Detailed Report

4

Experiments

4.1 Introduction

INTRUSION Detection and Prevention Systems (IDPS) are recommended and used most commonly to manage the security of the grid computing infrastructure. They have significantly improved the security of the traditional networks but when deployed in grid computing networks, they struggle to work stable due to the dynamic nature of the grid. This chapter covers four types of experiments. The first testifies the stability of the GSOC by efficient detection of distributed attacks and minimization of security alerts. The second shows the behavior of the components of the GSOC under distributed attacks. The third shows the significance of the GSOC in blocking propagation of cross domain attacks using intelligent sharing of security alerts. The fourth type covers the optimization in detecting distributed attacks. The first two types of experiments are discussed in section 4.2, third type is covered in section 4.3, and fourth in section 4.4.

The most common cause of the failure of any Security Management System (SMS) is its tendency to be stable under intense attacks. When intense attacks are launched in the network the SMS generates many alerts internally and correlate the reported alerts to find complex attacks. The correlation consumes processing power and needs time to conclude the type of the attack. There are many possibilities for generating too many security alerts by the SMS. Some of them are badly configured network, poorly defined security rules, security policies are not in place, hardware limitation, necessary updates are not done on time, human negligence, and the most important are the real attacks. The generation of too many security alerts lead the SMS to either report false positives or become unstable. The attacks vary from attacker to attacker because the experienced attackers use automated tools to lunch powerful

attacks in short period of time. The motive of these attacks is to first make the SMS unstable and then launch the real attack. When the SMS become unstable then either it does not detect the attack or it detects the attacks after some delay. GSOC which is proposed in this thesis works stable even if the above mentioned possibilities are true. It is designed to continue work under worst case scenarios. The objective of the first type of experiments is to prove that GSOC is more stable and has the tendency to continue its working under intense attacks. GSOC detects the distributed attack in real time. In GSOC the events are collected and processed locally at each local site of network. This is the reason that in GSOC the attacks are detected very early. Different comparisons of GSOC with other SMS are shown in the remaining part of this chapter.

4.1.1 Stability Comparison of SMS

Figure 4.1 is the graphical view which shows the comparison of different SMS. It shows the degradation in performance when the high intensity of attack continues for long period of time on the SMS. For comparing the performance of each SMS one round of BF attack was perform on a machine using a list of passwords. Guess Who performed more than 8000 passwords attempts on one victim machine. The red line in figure 4.1 shows the number of attempts made by the attacker on the victim machine. To detect this attack three SMS namely GSOC, DSOC and OSSIM are placed in parallel. The detection rate of GSOC remains stable throughout the attack. The detection rate of DSOC remains stable until 35 minutes but after that it starts degrading its performance. The detection rate of OSSIM remains stable until 20 minutes but after that it also starts degrading its performance. Both the DSOC and OSSIM detects the attacks but when the attack continues for long period of time their performance becomes bad to worse. This performance degradation is very crucial if the attacker gets success in its attacks. Due to this reason the alerts are reported with the delay and the delay continues to increase as shown in the graph. In this case even the DSOC and OSSIM will detect it after a certain delay but this delay give a fair chance to the attacker to do some harmful activities successfully. Whereas in the case of GSOC it continue detecting the attacks with a negligible delay.

To explain the first type of experiments, Snort [36] which is the most well-known IDS was deployed in the grid network, where its operational behavior was observed. Figure 4.2 is the general view of the Grid'5000 (G5K) [87] network infrastructure. In all the experiments G5K has been used to test the GSOC design. Snort was deployed at the Rennes site of the G5K network.

4.1.2 Introduction to Grid'5000 (G5K) Network

The G5K is the biggest grid network infrastructure dedicated to scientific experiments in France. The backbone of the network is provided by the French National Telecommunication Network for Technology, Education and Research (RE-

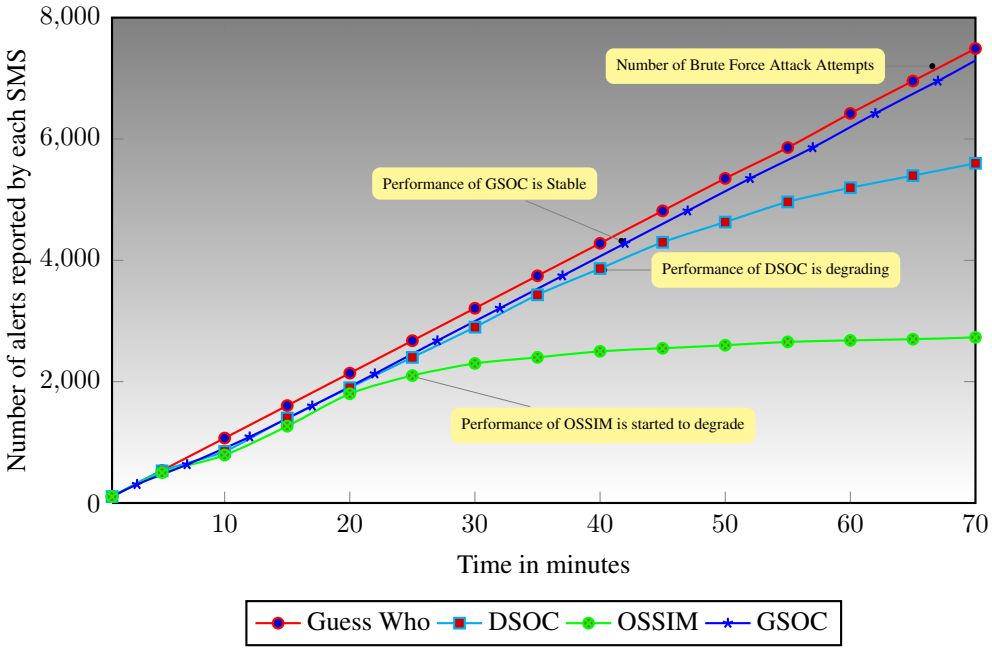


Figure 4.1: Stability of Different Security Management Systems

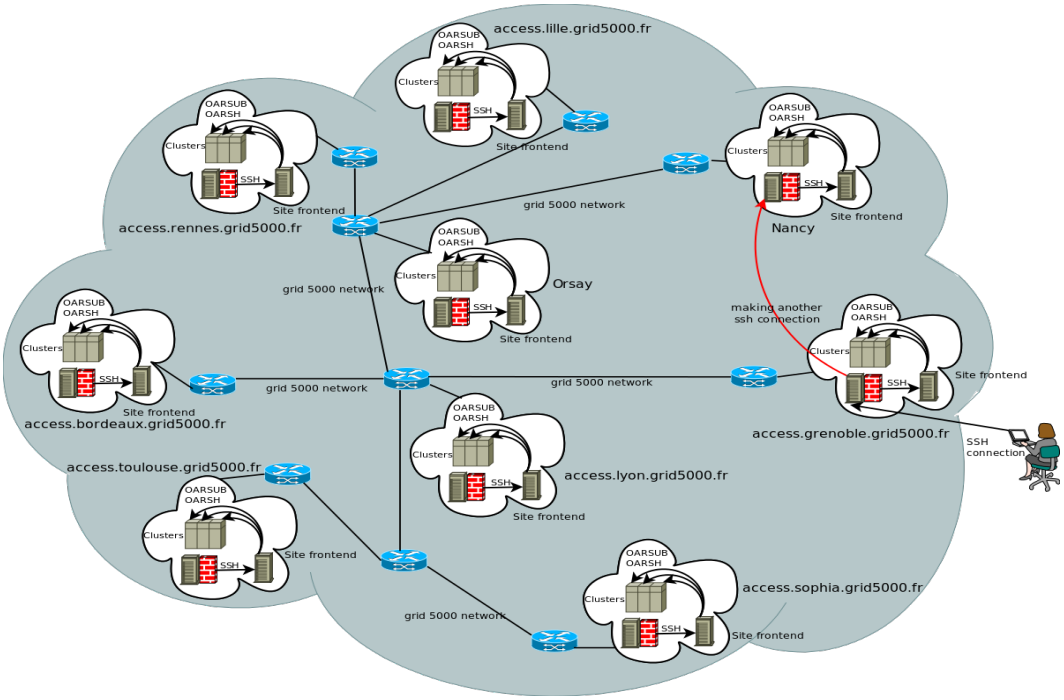


Figure 4.2: Grid'5000 General Overview

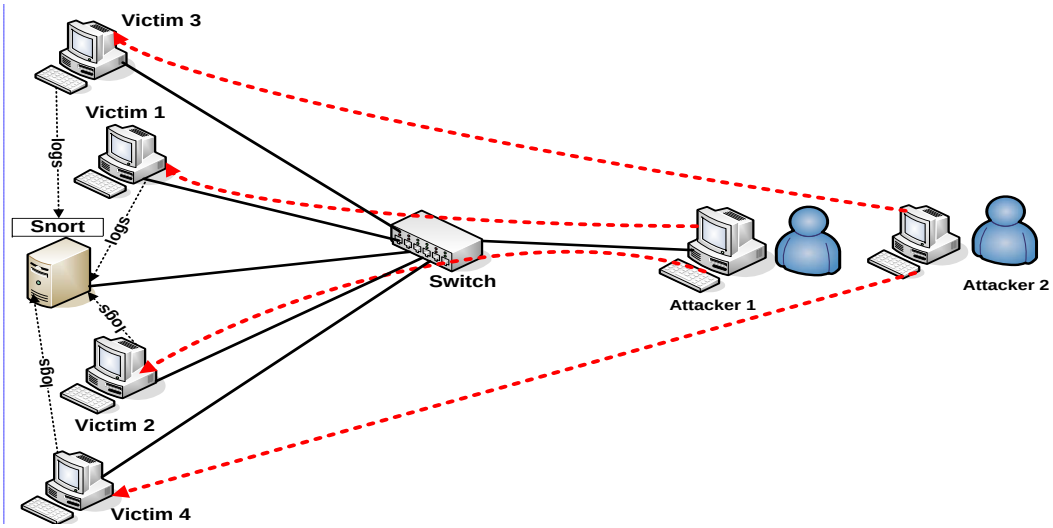


Figure 4.3: Snort Attack Scenario

NATER) [88]. The network is dispersed in nine cites of the France namely, Bordeaux, Grenoble, Lille, Lyon, Nancy, Orsay, Rennes, Sophia-Antipolis, and Toulouse as shown in figure 4.2 . Recently G5K has extended its network to the international partners and deployed one site in Porto Alegre at Brazil in 2011 and another at Luxembourg. The G5K uses the series of AMD Opteron and Intel Xeon processors having 10 Gb/s of bandwidth available between the sites using dark fiber. The infrastructure uses Myrinet and Infiniband communication links to minimize the bottlenecks between the G5K sites [87].

4.1.3 Snort Attack Detection in G5K

Snort uses a simple, lightweight rules description language that is flexible and quite powerful. It works by loading malicious traffic patterns called rules which help snort to identify the malicious traffic in the network. Snort only looks for what it is configured to detect and requires diligence updating by the security managers. Few attacks were performed on Snort in the G5K infrastructure. The Snort has generated huge amount of alerts. The number of alerts generated by the snort are plotted in graphs and explained in the subsections below. These high number of alerts are very critical because the security manager cannot find other attacks which could be more harmful in real time.

4.1.4 Snort under Brute Force (BF) Attack

When BF was lunched using THC Hydra [59] by two attackers (Attacker 1 and Attacker 2).

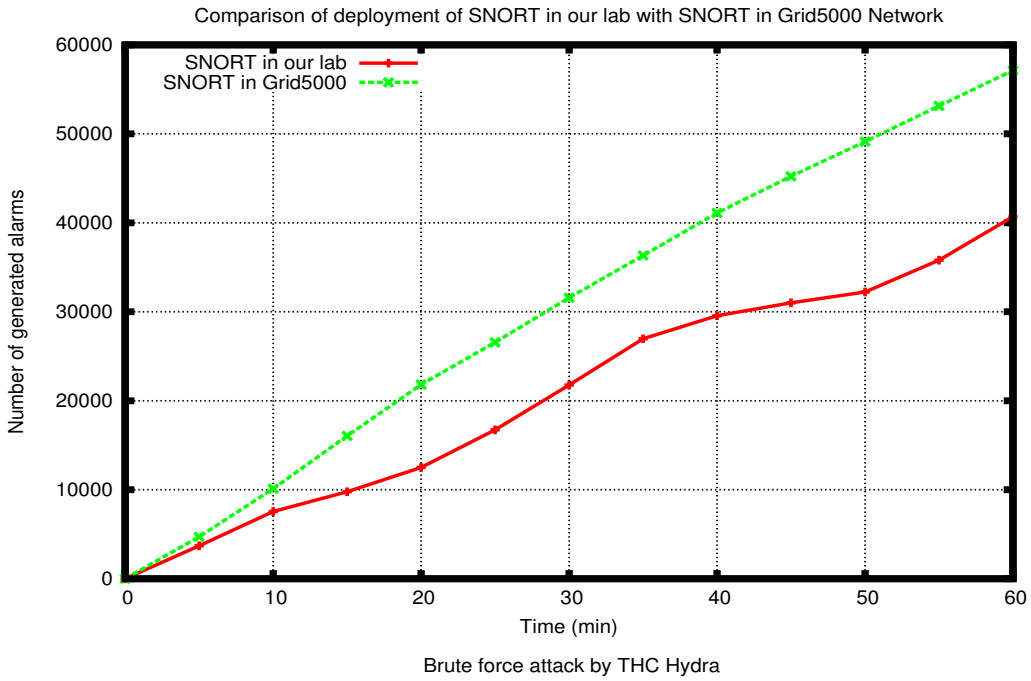


Figure 4.4: Snort under Brute Force Attack (BF)

Figure 4.3 illustrates the internal view of the attack scenario performed in G5K at the Rennes site. Attacker 1 targets victim machine 1 & 2 whereas Attacker 2 targets victim machine 3 & 4. Each attacker uses four instances of attacks and a total of eight together. All the four victim machines are configured to forward their logs to the Snort which is responsible for monitoring the security of a site. The rule for snort to detect these attacks is configured as,
Alert tcp any any -> IP ADDRESS PORT NUMBER (msg:"SSH"; flow: stateless; flag:S+)
This rule logs all the ssh connection messages from any source. When the attack was launched by two attackers, the victim machines logs and forwards all the connection failures attempts to the Snort. The result of the logs received and processed by the snort are shown in figure 4.4. Similar test was also repeated in the lab by making the same scenario shown in figure 4.3.

4.1.5 Snort Behavior under Ping of Death Attack

Using the same scenario explained in figure 4.3, Ping of Death (PoD) attack has been used to target the victim machines. The victim machines 1 to 4 start sending their logs to the Snort. The commands used by the attackers are the following:
Attacker 1: ping -i 0.5 -s 65507 IP Address of the Victim 1 & 2

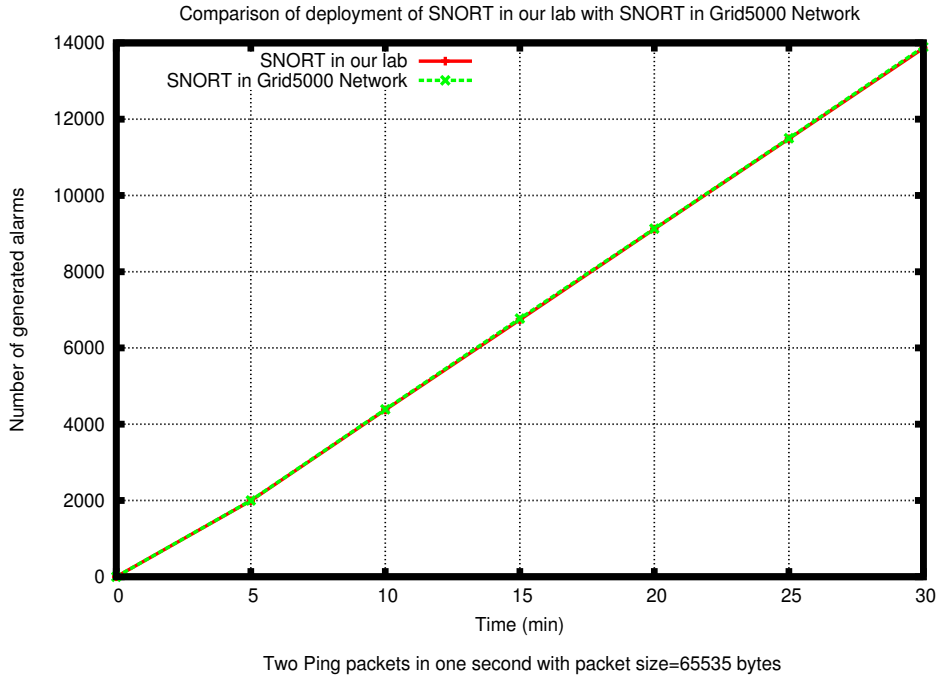


Figure 4.5: Snort Under Ping of Death Attack (PoD)

Attacker 2: ping -i 0.5 -s 65507 IP Address of the Victim 3 & 4
For Snort to generate alerts, a rule should be written to capture these ICMP packets. Using these commands one attacker sends 2 ICMP packets in one second having the size of 65507 bytes each. The rule is as follows:
alert ip any any -> any any (fragbits:!D; msg: "ICMP packet")
This rule detects any packet coming from any source towards any target machine. Snort marks that detected packet with the tag "ICMP packet" and saves it in the logs permanently. The same attacks has been repeated in G5K and in our lab. Both the attack cases of Snort shows that it has generated many alert messages for the same type of attack shown in figure 4.5.

4.2 Comparison of the Efficiency of Attack Detection

The objective of the experiments is to show how to efficiently minimize the number of security alerts when the site is under intense distributed or a combination of different attacks. This minimization of alerts helps the GSOC to work stable. It helps to reduce the alert processing time and bandwidth consumption which has direct impact on the performance of the GSOC which is shown in experiments. Minimization

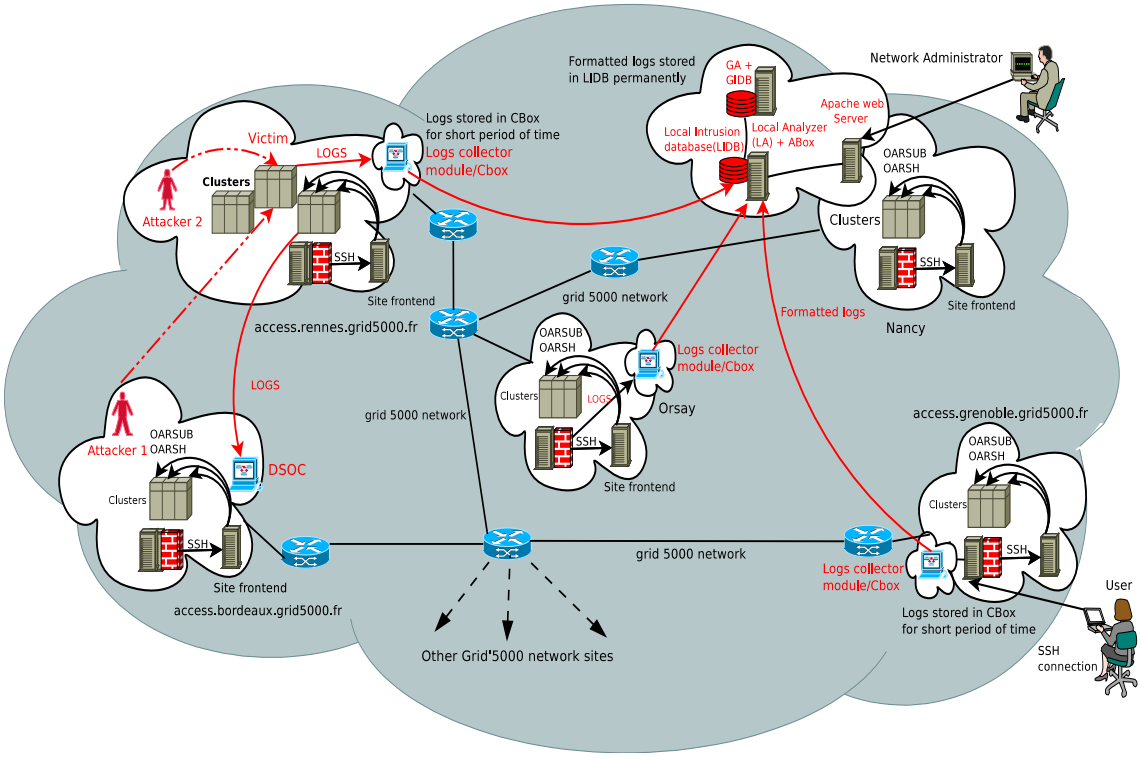


Figure 4.6: GSOC in Grid'5000 Network

of alerts is done using a correlation technique that keeps all the necessary information with the alert. In this section the comparison of the GSOC with the DSOC and the OSSIM under brute force and DDoS attacks is discussed. Due to the destructive nature of DDoS attacks the most intensive DDoS attacks were performed in our lab as we do not want to halt Grid'5000 network with our attacks. To calculate the efficiency, the number of alerts generated by the GSOC, DSOC and OSSIM in one hour is taken as a parameter. During the experiments on GSOC and DSOC some of the logs sent from the victim machine to the CBox machine were dropped due to network congestion because the UDP protocol had been used for sending and receiving the logs via rsyslog.

4.2.1 Behavior of GSOC Components under Multiple Attacks

In this section the GSOC capability to detect different attacks has been tested. In order to detect distributed attacks, the process of handling the alert by CBox and LA has been explained. Figure 4.6 is the reference of all the attacks explained in the tables. Table 4.1, table 4.2 and table 4.3 are divided into four columns. The first column contains the attack description. The second column displays the attack detection. The

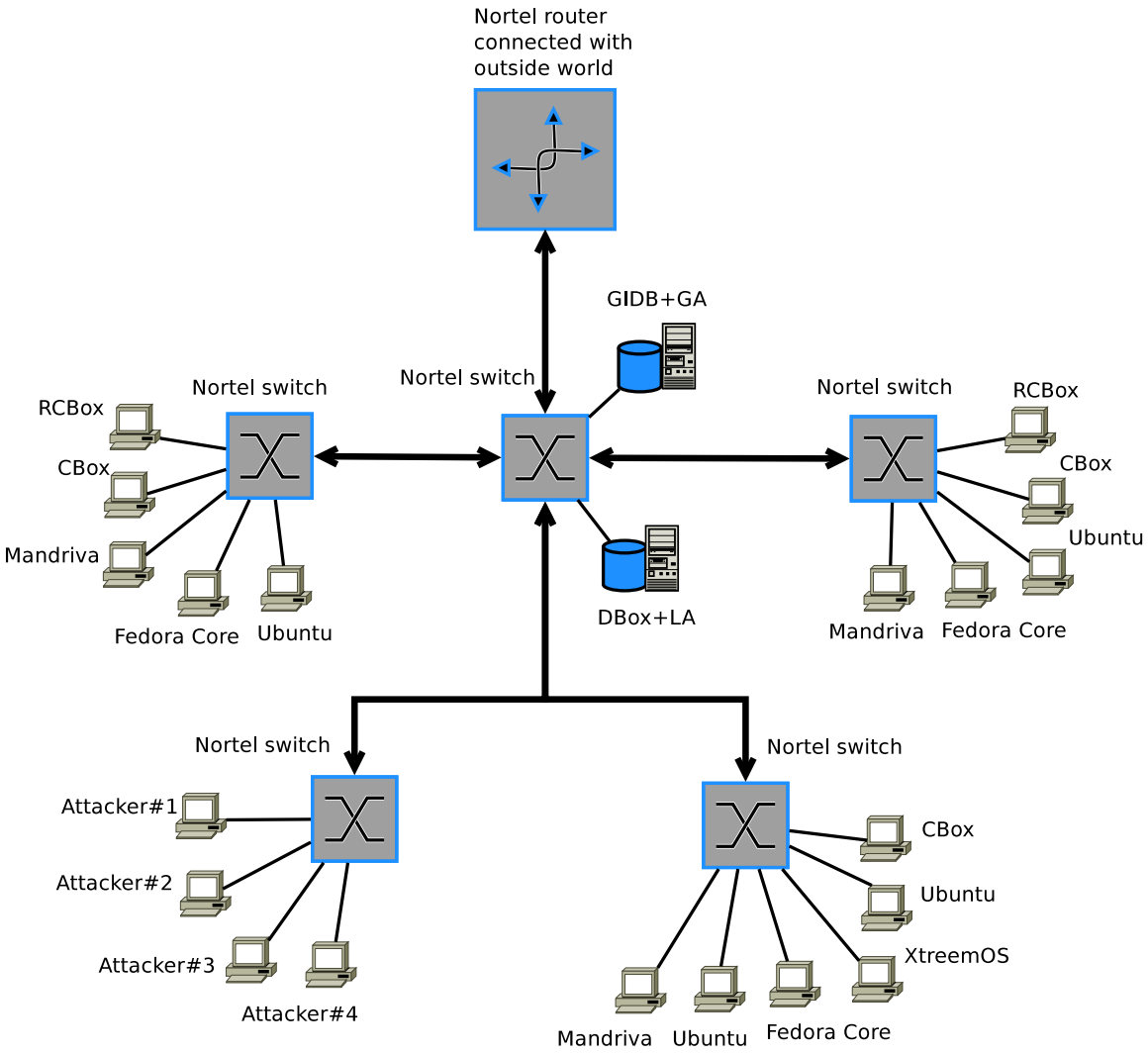


Figure 4.7: Lab Network Overview

third column shows the status on the CBox which is running in one local site. The fourth column points out the behavior of the LA which is controlling multiple CBoxes. The experiments have been performed in the laboratory of computer science located in the city of Montbéliard and in the Grid'5000 network [87]. In the lab 22 machines are used which are shown in the figure 4.7. The systems configuration is as follows:

Victim, attacker and CBox: CPU 2.4 GHz having 512 MB of RAM.

LA: CPU 2.66 GHz having 3 GB of RAM.

32-bit version of OSSIM was deployed on 1.5 GHZ processor having 512 MB of RAM.

In Grid'5000 (G5K) Network,

Victim: CPU 2.33 GHz 2 cores having 8 GB of RAM.

Attacker: CPU 2.33 GHz 2 cores having 8 GB of RAM.

CBox and LA: CPU 2.0 GHz 1 core each having RAM 2 GB of RAM.

In the G5K, Rennes was used for attackers, the CBox and victim machines. Nancy for the LA and Bordeaux for the DSOC. Please refer to figure 4.6 for details.

4.2.2 Description of the Lab Network

The network of the Université de Franche-Comté is very complex because the main campus is located in Besançon having two remote campuses which are located at Belfort and Montbéliard cities. The figure 4.7 is the overview of the network infrastructure of the Montbéliard campus. The Nortel router is the gateway which provides connectivity from the main campus via fiber optic link. The network is distributed in every department using Nortel BPS 2000 switches. All the switches are connected via each other with the fiber optic links. The user machines are connected using Cat5 cables in the labs. The network is composed of many components such as FTP, Web and Storage servers that are placed in each department. For simplicity only the machines that are used for the experiments are drawn. The machines are distributed in our lab, which uses multiple type of operating systems.

4.2.3 GSOC, DSOC and OSSIM under Brute Force (BF) Attack

For launching the brute force attack, THC hydra [59] was used with a password file of 8048 passwords. The required password was placed at the 8048th place in the password file of Attacker 1 and no correct password was provided to Attacker 2. In this test two attackers (Attacker 1 and Attacker 2) are performing the attack on a target machine called victim as show in figure 4.6. This diagram shows the Grid'5000 network infrastructure. The victim is a machine located at the Rennes site. There are different clusters having many nodes at the Rennes site, the victim machine is a node we reserved for the experiments. Attacker 1 launches an attack from the Bordeaux site and Attacker 2 launches an attack from the Rennes site. The logs generated by the victim machines are forwarded to the CBox placed in another node at the Rennes site.

Table 4.1: Flooding Attacks Detection Capabilities in the GSOC

Flood pollution attacks Description	**Detection**	**Action on CBox**	**Action on LA**
Attacker floods the GSOC sensor with Apsend [89], followed by a BF attack (with THC Hydra [59])	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	No alarm generated for a strong BF attack, as there is only one attacker
Attacker floods the GSOC sensor with Apsend, followed by a BF attack (with Guess Who [60])	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	No alarm generated for a strong BF attack, as there is only one attacker
Attacker floods the GSOC sensor with Apsend followed by a BF attack (with Medusa [90])	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	No alarm generated for a strong BF attack, as there is only one attacker
Attacker 1 floods the GSOC sensor with 2000 pings generated by Apsend, followed by a brute force attack from THC Hydra. Attacker 2 generates a BF attack by Medusa followed by Guess Who	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	The LA reports a strong BF attack, pointing out the victim sensor and the source machines of attackers 1 & 2 responsible for the attack
Attacker 1 floods GSOC sensor with 4000 SYN packets generated by Apsend, followed by a BF attack from THC Hydra. Attacker 2 generates a BF attack with Medusa followed by Guess Who	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	The LA reports a strong BF attack, pointing out the victim sensor and the source machines of attackers 1 & 2 responsible for the attack

Table 4.2: Brute Force (BF) Attack Detection Capabilities in GSOC

BF attack Description	Detection	Action on CBox	Action on LA
Attacker launches a brute force attack on a sensor with Medusa	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	No alarm generated for a strong BF attack, as there is only one attacker
Attacker launches a BF attack on a sensor with Guess Who	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	No alarm generated for a strong BF attack, as there is only one attacker
Attacker 1 launches a BF attack on a sensor with THC Hydra and Attacker 2 generates a BF attack with Guess Who on the same sensor	YES	The CBox detects a weak BF attack on the victim sensor and sends formatted alerts to the LA	Alarm for a strong BF attack is generated, pointing out the victim sensor and the source machines responsible for the attack

These logs are minimized by basic correlation at the CBox and then sent to the LA for advanced correlation which further minimizes the logs and generates few alarms. The alarms which are displayed on the GUI by using the apache web server are stored in the DBox.

The behavior of the GSOC can be seen in figures 4.8 and 4.9. Each CBox correlates the local site alerts of the AD and reports to the GUI as a weak attack as shown in figure 4.10. The reported alerts at each CBox are forwarded to the ABox for advanced correlation. The alerts received by the ABox from different CBoxes are finally reported as the strong attack alarm as shown in figure 4.11). The alerts reported at the CBox and alarms at the ABox contain all the necessary information which includes the IP address of the sources, the start and end time of the attack which is equal to the elapsed time of one minute, the user's name (Attacker 1 or Attacker 2) by whom the attack has been launched. They also provide the target IP addresses and the number of attempts (count) made by each attacker. This information is very helpful for the security manager to stop these attacks from expending.

The behavior of the DSOC in figures 4.8 and 4.9 shows that the DSOC has generated one alert for every password attempt. This shows that in one hour the security manager has received more than 16K alerts in our lab and more than 22K in the Grid'5000 network from one victim, for scenario see figure 4.6.

Table 4.3: Distributed Denial of Service (DDoS) Attack Detection Capabilities in GSOC

DoS and DDoS Attacks Description	**Detection**	**Action on CBox**	**Action on LA**
Attacker generates a ping of death attack (PoD) with a packet size of 65535 bytes with time interval is 0.2 second	YES	DoS is detected, pointing out the attacker's machines. After the detection a formatted alert is sent to the LA	No alarm generated for the DDoS, since there is only one attacker
Attacker 1 and Attacker 2 generate a PoD attack with a packet size of 65535 bytes with time interval is 0.2 second	YES	DoS is detected, pointing out the attackers machines. After the detection, a formatted alert is sent to the LA	Alarm for a DDoS attack is generated, pointing out the attackers machines responsible for the attack
Attacker launches a Slowloris attack on an Apache web server	YES	DoS is detected, pointing out the attacker's machines. After the detection, a formatted alert is sent to the LA	No alarm generated for the DDoS, since there is only one attacker
Slowloris attack on an Apache web server, followed by a PoD attack with a packet size of 65535 bytes, followed by a BF attack (by THC Hydra)	YES	Two different types of DoS attacks (Slowloris and PoD) and a Weak BF attack are detected, pointing out the attacker's machine. After each attack detection, a formatted alert is sent to the LA	No alarm generated for the DDoS, since there is only one attacker
Slowloris attack on an Apache web server in parallel with a PoD attack with a packet size of 65535 bytes and a BF attack (by THC Hydra) from one attacker	YES	Two different types of DoS attacks (Slowloris and PoD) and weak BF attacks are detected, pointing out the attacker's machine. After each attack detection, a formatted alert is sent to the LA	No alarm generated for the DDoS, since there is only one attacker

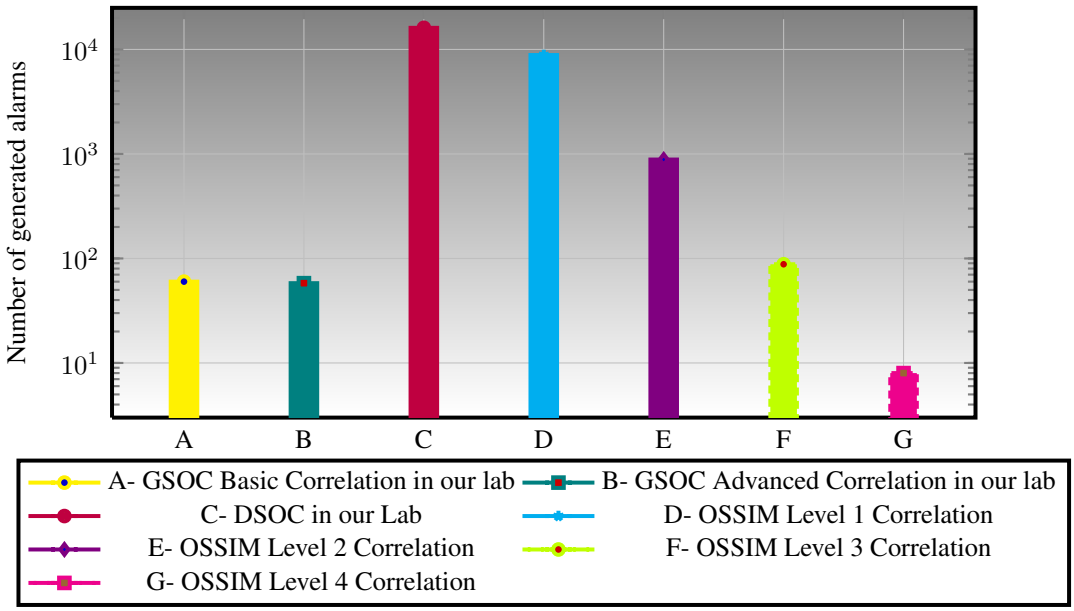


Figure 4.8: GSOC Comparison with OSSIM and DSOC under Brute Force Attack in Our Lab

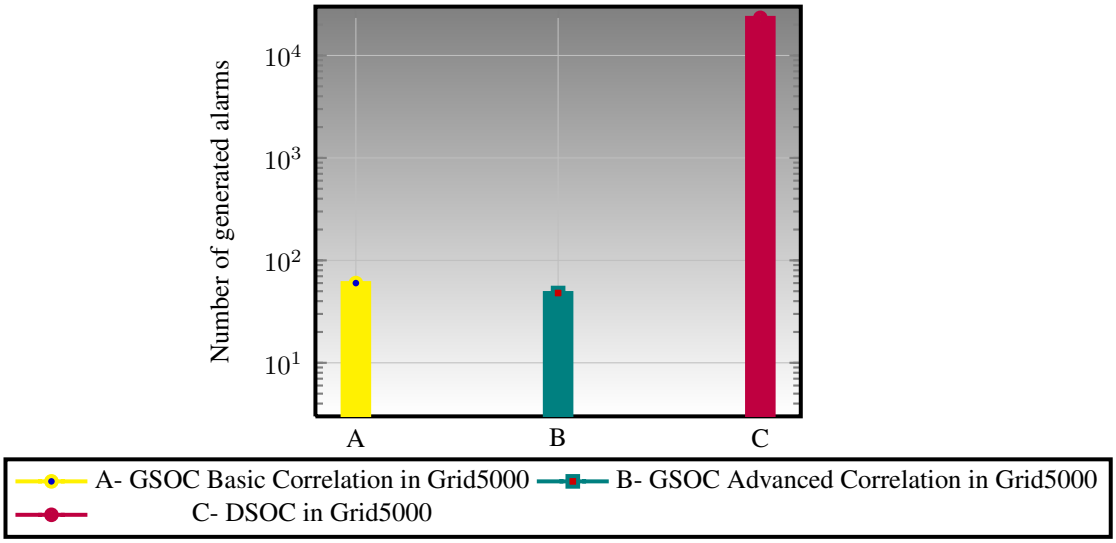


Figure 4.9: Deployment of GSOC and DSOC in Grid'5000 Network under Brute Force Attack

Start Time	End Time	Type	Target	Source => Target User / count
16:57:06	16:58:06	Weak attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_1 / 932
16:58:07	16:59:07	Weak attack: Authentication failure	131.254.202.191	131.254.202.190 => attacker_1 / 464
16:59:09	17:00:09	Weak attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_2 / 178
17:00:10	17:01:10	Weak attack: Authentication failure	131.254.202.191	131.254.202.190 => attacker_1 / 4270
17:01:12	17:02:12	Weak attack: Authentication failure	131.254.202.191	131.254.202.190 => attacker_1 / 1490
17:02:13	17:03:13	Weak attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_2 / 136
17:03:14	17:04:14	Weak attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_2 / 144

Figure 4.10: GSOC GUI: Weak Alert Reported to CBox

Start Time	End Time	Type	Target	Source => Target User / count
16:54:03	16:55:03	Strong attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_1 / 556
16:55:04	16:56:04	Strong attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_1 / 602
16:56:05	16:57:05	Strong attack: Authentication failure	131.254.202.191	131.254.202.190 => attacker_1 / 840
16:57:06	16:58:06	Strong attack: Authentication failure	131.254.202.191	131.254.202.190 => attacker_1 / 514
16:59:09	17:00:09	Strong attack: Authentication failure	131.254.202.191	131.254.202.190 => attacker_1 / 1110
17:00:10	17:01:10	Strong attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_2 / 312
17:01:12	17:02:12	Strong attack: Authentication failure	131.254.202.191	131.254.202.172 => attacker_2 / 144

Figure 4.11: GSOC GUI: Strong Alert Reported to LA

The behavior of the OSSIM in figure 4.8 shows that level 1 correlation passes all the authentication failure alerts to the GUI (refer to the code of level 1 correlation). Level 2 correlation starts working when level 1 correlation finishes after receiving one authentication failure. Level 2 correlation covers two possibilities first the authentication success which could occur after one authentication failure in level 2 correlation. This means that there is a total of two authentication failures, one from level 1 correlation and other from level 2 correlation. Second it receives authentication failures for the period of 10 seconds defined in the "time_out" field. If one authentication successful event is received within 20 seconds then level 2 correlation is finished. Level 2 correlation merges multiple occurrences of authentication failures and generates only one alert (refer to the code of level 2 correlation). If no authentication successful event is received then it will pass the information to level 3 correlation.

Level 3 correlation starts working when level 2 correlation finishes after receiving 20 or more occurrences in 10 seconds. The level 3 correlation also covers two possibilities: first, the authentication success which could occur after one authentication failure in level 3. This means that there is a total of more than 20 authentication failures, one from level 1 correlation, 20 from level 2 correlation. Second it receives authentication failures for the period of 40 seconds defined in the "time_out" field. If one authentication successful event is received within 40 seconds then level 3 correlation is finished. Level 3 Correlation merges multiple occurrences of authentication failures and generates only one alert (refer to the code of level 3 correlation). If no authentication successful event is received then it will pass the information to level 4 correlation.

Level 4 correlation starts working when level 3 correlation finishes after receiving 100 or more occurrences in 40 seconds. Level 4 correlation also covers two possibilities: first, the authentication success which could occur after one authentication failure in level 4 correlation. This means that there is a total of more than 100 authentication failures, one from level 1 correlation, 20 from level 2 correlation and 100 from level 3 correlation. Second it receives authentication failures for the period of 300 seconds defined in the "time_out" field. If one authentication successful event is received within 300 seconds then level 4 correlation is finished. Level 4 correlation merges the 1000 occurrences of authentication failures and generates only one alert (refer to the code of level 4 correlation).

Code of Level 1 Correlation

```
1 % <directive id="500010" name="SSH Brute Force Attack Against ANY_IP" priority="5">
2 %   <rule type="detector" name="SSH Authentication failure" reliability="0"
3 %     occurrence="1" from="ANY" to="ANY" port_from="ANY" port_to="ANY"
4 %     plugin_id="4003" plugin_sid="1,2,3,4,5,6,9,10,12,13,14,15,16,20">
5 %   <rules>
```

Code of Level 2 Correlation

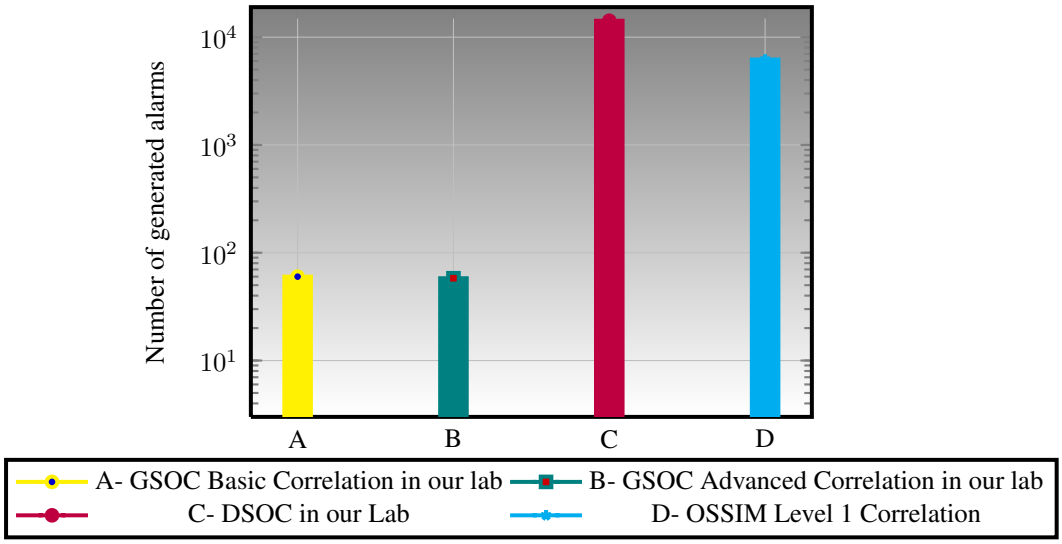


Figure 4.12: Comparison of GSOC with OSSIM and DSOC under Ping-of-Death Attack in Our Lab

```
<rule type="detector" name="SSH Successful Auth (After 1 failed)"
   reliability="1" occurrence="1" from="1:SRC_IP" to="1:DST_IP"
   port_from="ANY" time_out="10" port_to="ANY" plugin_id="4003" plugin_sid="7,8"/>
<rule type="detector" name="SSH Auth failure (10 times)"
   reliability="2" occurrence="20" from="1:SRC_IP"to="1:DST_IP"
   port_from="ANY" time_out="10" port_to="ANY"
   plugin_id="4003" plugin_sid="1,2,3,4,5,6,9,10,12,13,14,15,16,20" sticky="true">
 <rules>
```

Code of Level 3 Correlation

```
<rule type="detector" name="SSH Successful Auth (After 1 failed)"
   reliability="3" occurrence="1" from="1:SRC_IP" to="1:DST_IP"
   port_from="ANY" time_out="40" port_to="ANY" plugin_id="4003" plugin_sid="7,8"/>
<rule type="detector"
   name="SSH Auth failure (100 times)" reliability="4" occurrence="100"
   from="1:SRC_IP" to="1:DST_IP" port_from="ANY" time_out="40" port_to="ANY"
   plugin_id="4003" plugin_sid="1,2,3,4,5,6,9,10,12,13,14,15,16,20" sticky="true"/>
</rules>
```

Code of Level 4 Correlation

```
<rule type="detector" name="SSH Successful Authentication (After 1 failed)"
   reliability="5" occurrence="1" from="1:SRC_IP" to="1:DST_IP"
   port_from="ANY" time_out="300" port_to="ANY" plugin_id="4003" plugin_sid="7,8"/>
   <rule type="detector" name="SSH Authentication failure (1000 times)"
   reliability="6" occurrence="1000" from="1:SRC_IP" to="1:DST_IP"
   port_from="ANY" time_out="300" port_to="ANY"
   plugin_id="4003" plugin_sid="1,2,3,4,5,6,9,10,12,13,14,15,16,20" sticky="true"/>
</rule>
</rules>
```

4.2.4 GSOC, DSOC and OSSIM under Ping of Death (PoD) Attack

This is a ping-of-death(PoD) attack scenario for the GSOC, DSOC and OSSIM as shown in figure 4.6. The attack scenario is the same as discussed in subsection

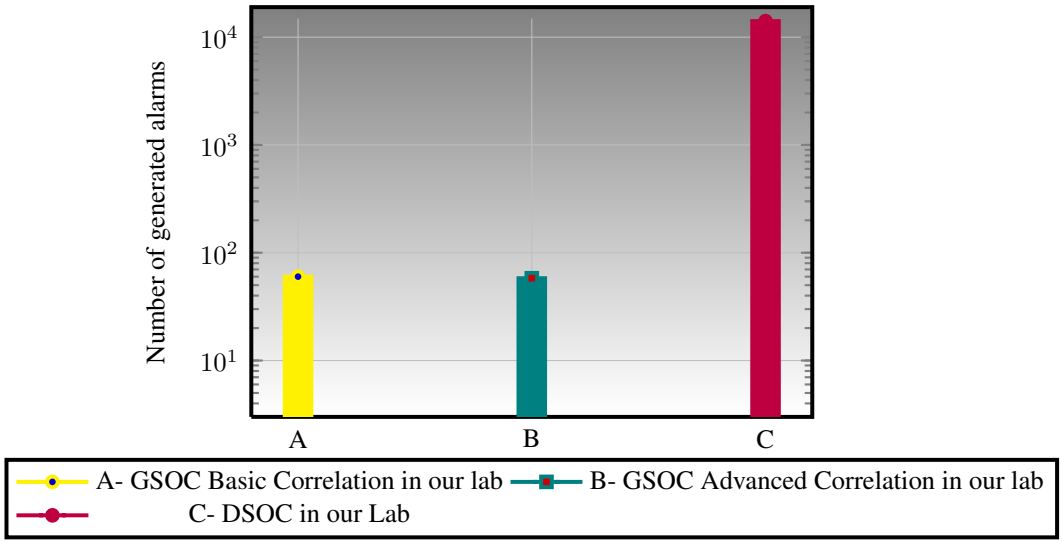


Figure 4.13: Deployment of GSOC and DSOC in Grid'5000 Network under Ping-of-Death Attack

4.2.3. The hping commands were used to launch PoD attacks. The hping is a free packet generator and analyzer for TCP/IP [80]. The commands below send a two ping request per second, having the size of a 65495-byte packet to the victim machine.
Attacker 1: hping3 -i u500000 -d 65495 IP Address of the Victim
Attacker 2: hping3 -i u500000 -d 65495 IP Address of the Victim

The motive of the tests is to show the performance of the GSOC, DSOC and OSSIM when attackers are generating huge traffic that would lead to DoS/DDoS attacks. These attacks are used to camouflage the real attacks. Hping can be used to launch very sophisticated attacks which can halt the operation of a network. I have not performed very critical attacks because I do not want to stop the operation of any network. In order to detect a DDoS/DoS attack in the GSOC, ping packets bigger than 85 bytes are discarded. If any packet bigger than 85 bytes is received by the CBox the Iptable rules are used to log an alert . In the code of the GSOC when the CBox script executes the iptables, rules are added automatically. These iptable rules generate kernel warnings which means that each ICMP packet greater than 85 bytes will be reported. The results can be seen in figures 4.12 and 4.13. The iptable rules that are used are as follows:
Iptables -A INPUT -d0/0 -s0/0 -p icmp -m length -length 85: -j LOG -log-prefix "PING OF DEATH"
Iptables -A Iptables -A INPUT -d0/0 -s0/0 -p icmp -m length -length 85: -j DROP

The behavior of the DSOC shows that it generates one alert per hping request. That means two alerts per second from one attacker. This shows that in one hour the network administrator has received more than 14K alerts from one victim, that can be

Table 4.4: Performance Comparison of GSOC, DSOC and OSSIM

Name of Tool	**Start Time** (t_s)	**End Time** (t_e)	**Total Time** (t_t)
THC HYDRA	12H:17min	13H:32min	1H:15min
GSOC	12H:17min	13H:38min	1H:21min
DSOC	12H:17min	13H:50min	1H:33min
OSSIM	12H:17min	17H:45min	5H:28min

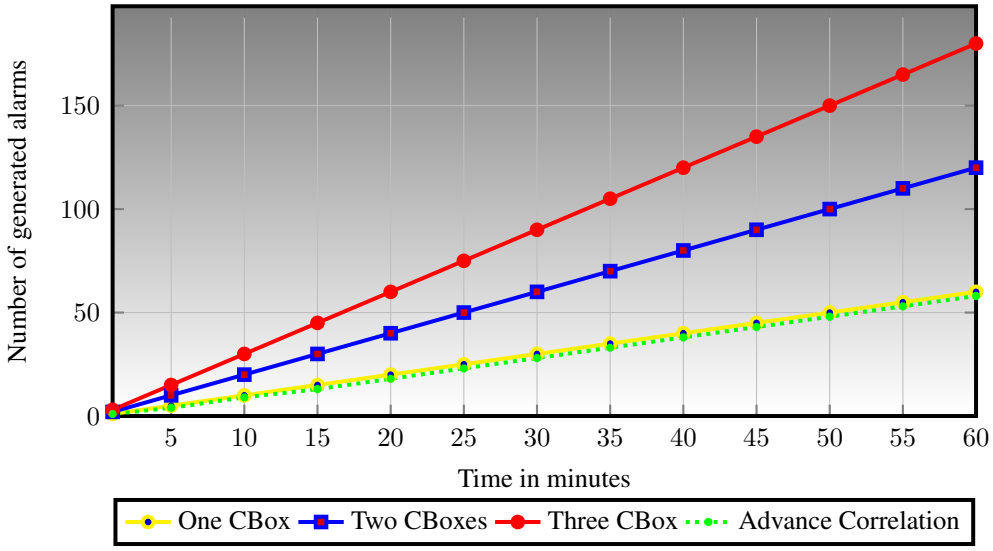


Figure 4.14: Correlation of Security Alerts Coming from Different CBoxes

seen in figures 4.12 and 4.13.

The OSSIM was not deployed in G5K because the OSSIM support is not available in their network. According to the hping command, from two attackers OSSIM has to generate 4 alerts in one second and a total of 14400 alerts in one hour. But due to the slow processing of alerts OSSIM lacks in spontaneous reporting which can be seen in figure 4.12.

Discussion :
The objective of the experiments was to minimize the number of security alerts when the sites are under intense distributed or a combination of attacks while lowering down the reporting delay. Table 4.4 shows the performance of the GSOC, DSOC & OSSIM and how quickly they detect and report security alerts. The THC Hydra took 1

Table 4.5: Approximate Database Utilization of GSOC, DSOC and OSSIM

Name of Tool	Brute Force Attack	Denial of Service Attack
GSOC BC	28KB	26KB
GSOC AC	14KB	11KB
DSOC	3819KB	3370KB
OSSIM	2098KB	1471KB

hour 15 min to complete the list of 8048 passwords. The GSOC detects all the failure attempts and completes reporting within 1 hour 21 min. The DSOC detects all the failure attempts and completes reporting within 1 hour 33 min. The OSSIM detects all the failure attempts and completes reporting within 5 hour 28 min. The tests were started simultaneously, the results show that GSOC processing capability is much better than that of the DSOC and that of the OSSIM. Therefore the delay in reporting is directly proportional to the risk of network compromise. Although in GSOC there is a delay of 60 seconds while the alert moves from CBox to LA as explained in section 3.3, but this delay is low as compared to the DSOC and the OSSIM.

Figure 4.14 shows the alert correlation. Each CBox generates certain number of alerts depending on the type of the attack. Each CBox then sends these alerts to the ABox for advance correlation. The ABox receives all the alerts and merges them according to the similarity of the the type of the attacks launched by the same or multiple sources. The yellow line shows the behavior when one CBox is deployed. The blue line shows the behavior when two CBoxes are deployed. The red line shows the behavior when three CBoxes are deployed. The green dotted line which denotes the behavior of the advance correlation will remain same when one, two or three CBoxes are deployed because ABox does time correlation and it generates only one alert in one minute. If one or multiple sources are involved in attacking all the three CBoxes. It will be detected at advance correlation and reports for the distributed attack to the security manager. This helps in identifying the actual root cause of the attack whereas with basic correlation this was not possible. Details are discussed in the tables 4.1, 4.2 and 4.3.

Table 4.5 shows the database utilization of GSOC, DSOC and OSSIM during the BF and DoS attack. The BF was launched by two attackers using the dictionary of passwords which contains 8048 passwords. Two CBoxes sends their logs to the LA which detects strong BF attack. In the DoS attack two pings of 65495-byte per

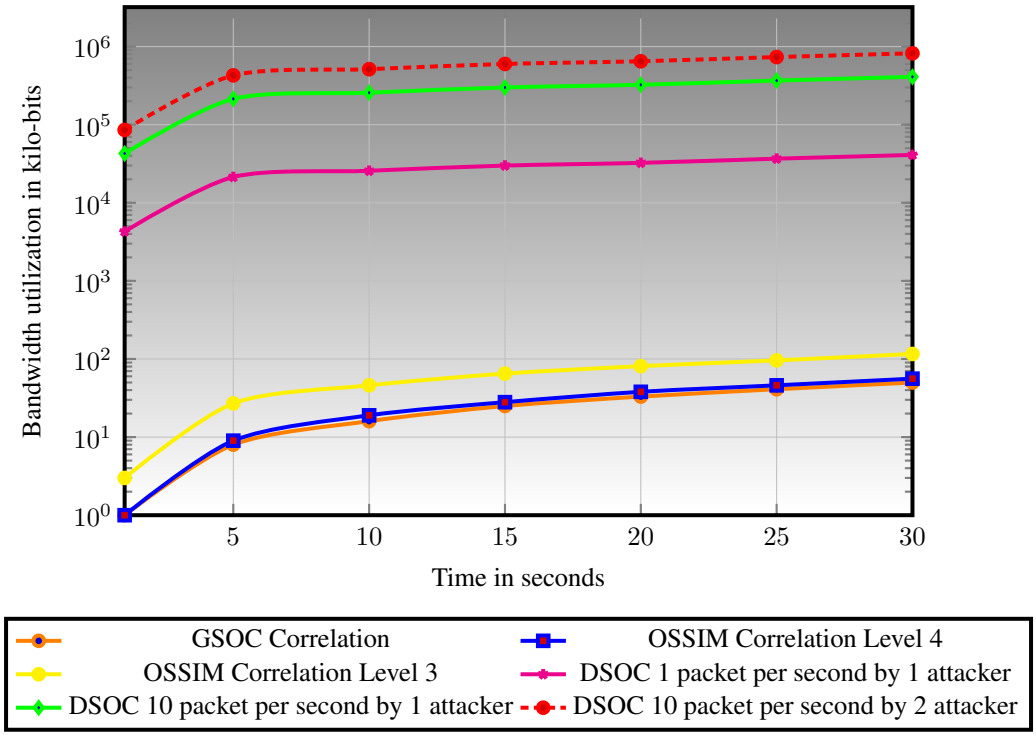


Figure 4.15: Bandwidth Utilization under DoS Attack

second were sent at the victim machines. Two CBoxes sent their logs to the LA for the detection of DoS attack. These are the approximate size of the alerts which are stored in a database. Whereas in a running system there are many other parameters which are stored with the alerts like sensor uptime, operating system information, asset values, priorities, etc. which are not considered here.

Figure 4.15 shows bandwidth utilization of GSOC, DSOC and OSSIM during different frequencies of DoS attacks. The actual attack lasts for more than one hour with flood mode option enabled using hping but here only the first 30 seconds are taken for showing the difference. In this test 1 packet and 10 packets were sent per second to the victim machines having the packet size of 65100 bytes. The Snort uses the rule defined as **alert ip any any ->any any (fragbits:!D; msg: "ICMP packet"**). The "any any" option allows snort to sniff any packet from any source to any destination. The "->" operator sets the direction to sniff the packets. The option "fragbits:!D" generate alert for every IP packet that does not have the fragment bit set. The option "msg" stores the "ICMP packet" tag and logs it in the snort database and forwards to the CBox. In the experiments the Maximum Transmission Unit (MTU) was 1500 bytes. Therefore when the ICMP packet of 65100 bytes was set to send on the network, it was fragmented into 1480 bytes of packets while 20 bytes were reserved for the IP

header. For transmitting 65100 bytes on the network, fragmentation was used. The first 43 packet of 1480 bytes gives the total of 63640 bytes and a last packet (44th) of 1460 bytes + 20 bytes of IP header. This rule generated 44 alerts for every fragment of the packet because Snort was working in a sniffer mode and forwards every tag of ICMP packet to the CBox.

The first four graphs (brown, blue, yellow and pink) which represents GSOC correlation, OSSIM correlation level 4, OSSIM correlation level 3 and DSOC were generated when one attacker sent one packet in one second to the victim machine. The fifth (green) graphs shows the behavior under flood mode this means 10 packets were sent from one attacker to the victim machine in one second. The sixth (red) graph shows the behavior under flood mode this means 10 packets were sent from two attackers to the victim machine in one second.

4.3 Blocking Propagation of Cross Domain Attacks

This section compares the behavior of two types of security management systems. The first one is developed for traditional computer network, but could be deployed in grid computing networks; the second one is developed for grid computing networks. The Distributed Security Operation Center (DSOC), which while developed for traditional computer networks has been used for executing different tests is classified in figure 2.4 under the category “Security solutions for traditional networks deployable in grid computing networks." The DSOC will be used for representing other security solutions that exist in a similar category. Graphs 4.17 and 4.18 show the security alert rate in minutes. The graphs shows three attack details, namely those of Brute Force attacks (BF), Denial of Service attacks (DoS), and Distributed Denial of Service attacks (DDoS) launched on DSOC and on GSOC using multiple sites of the Grid’5000 (G5K) network.

4.3.1 Attack Scenario-I

Figure 4.16 consists of two parts. The upper part represents a simplified view of G5K network with CBox running at the Rennes site. The LA+ABox, LIDB, GA+GIDB are running at the Nancy site. The approved users are allowed access to these sites where they can reserve many nodes and perform their experiments. These users are also allowed to reserve any number of machines between nine other sites of the G5K.

The lower part is a simplified view of our lab network where two users from Machine 1 and 2 via ssh connection ① & ② are allowed to connect into the G5K network. Attackers 1 & 2 try to get access to machines 1 and 2 by launching brute force attacks using a dictionary of passwords that contains 8048 passwords. Attacker 2 was not successful on Machine 1 due to a strong password. Attacker 1 was successful and cracked the password of Machine 2 after 10 minutes which is found at the 5000th

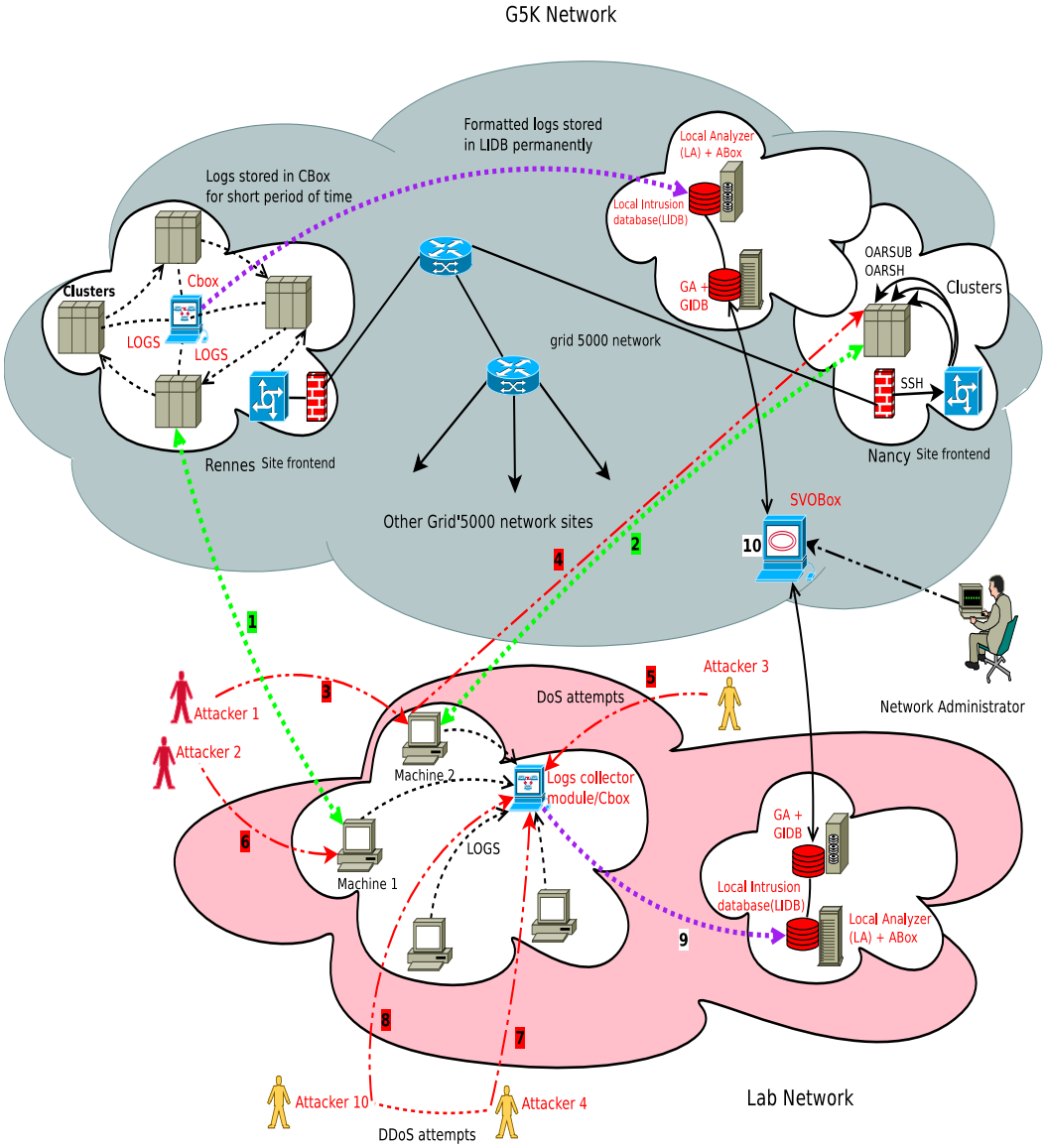


Figure 4.16: Stopping Propagation of Cross-Domain Attacks between Our Lab and G5K Network

location of the password file③. Attacker 1 using machine 2 is a threat within local and external networks which are connected together. Attacker 1 can perform malicious activities in G5K since the connection originated from the approved user machine. For the G5K network, Machine 1 was a trusted machine but is now compromised; ④ shows that the G5K network is accessible by Attacker 1. Attacker 1 can further launch more brute force attacks on other machines of the G5K. If successful, the results are very destructive.

4.3.2 Attack Scenario-II

Usually, an experienced attacker uses a combination of multiple attacks to hide their activities. The easiest scenario is to first launch a DoS attack from Attacker 3 that generates the alerts of DoS ⑤. These alerts are generated deliberately to attract the attention of the network administrator and fill the GUI of the security management system. After some time Attacker 2 from the other machine starts launching another attempt of brute force attacks ⑥ to crack the password of Machine 1. These attacks last for short periods of time and are restarted after some time.

4.3.3 Attack Scenario-III

A more complex scenario is to launch the DDoS from multiple machines (Attacker 4 to Attacker 6) by spoofing IP addresses ⑦ & ⑧. These attackers generate several alerts and because of the IP spoofing need more time from the administrator to detect the actual source of the attacks. These attacks cover two objectives: It overloads the network and its components so the legitimate users cannot access it and allows them to destabilize the security management systems. In this way, the malicious activities are not easily detectable and if they are detected, they will be reported very late to the administrator due to the high number of security alerts processing time.

By deploying GSOC between G5K and our lab network, attack types I, II & III can be blocked at very early stage. The CBox which is running in our network collects all unsuccessful authentication failures attempts and sends them to the LA ⑨. The LA correlates all the logs, generates a brute force attack attempt, and saves it in LIDB. The sharing mechanism of GSOC allows the lab network to share this information with the G5K network. The administrator of the G5K network has access to the brute force attempt alarm in our lab which includes the IP address, user ID, start time of attack, end time of attack, and total number of fail attempts ⑩. This information helps the network administrator of G5K to stop the access of that user. In this way an attack which is propagated from one administrative domain to another can be blocked.

Discussion :
In figure 4.17 the DSOC detects the attacks between 1 or 2 minutes and reports them

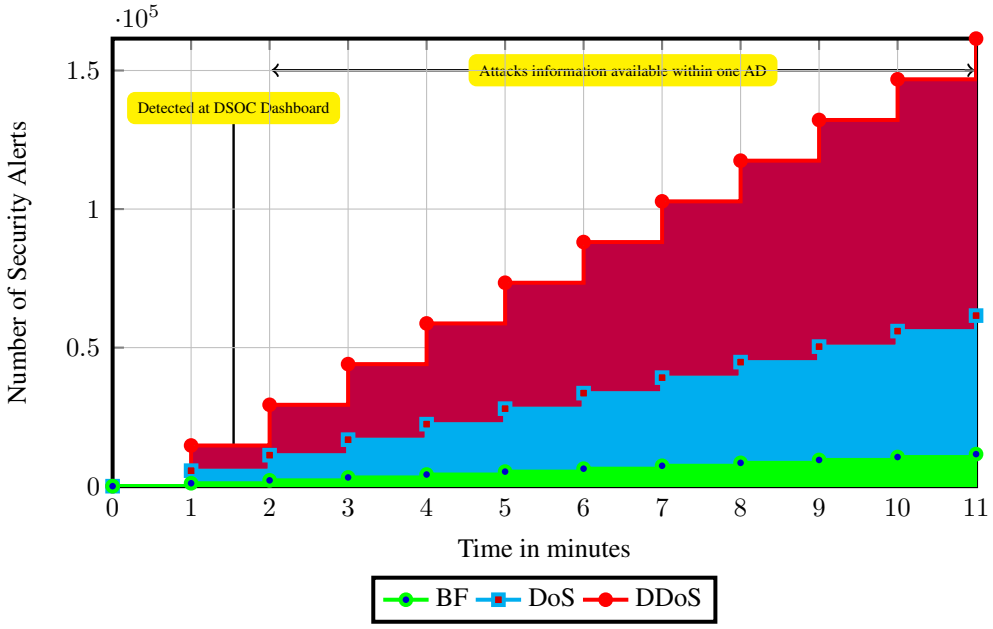


Figure 4.17: Detection Rate of BF, DoS and DDoS in Distributed Security Operation Center (DSOC)

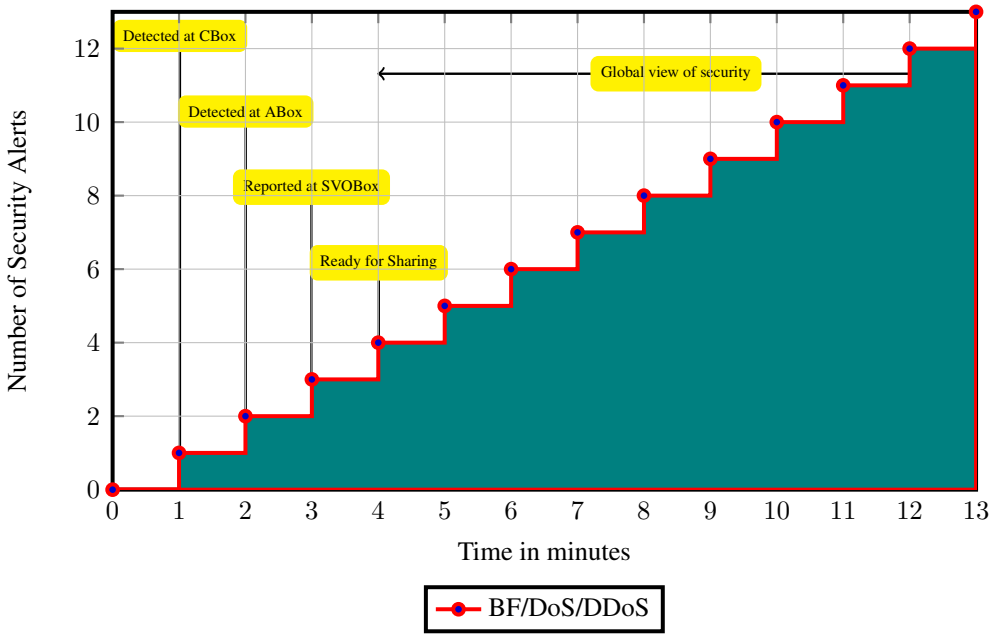


Figure 4.18: Detection Rate of BF, DoS and DDoS in Grid Security Operation Center (GSOC)

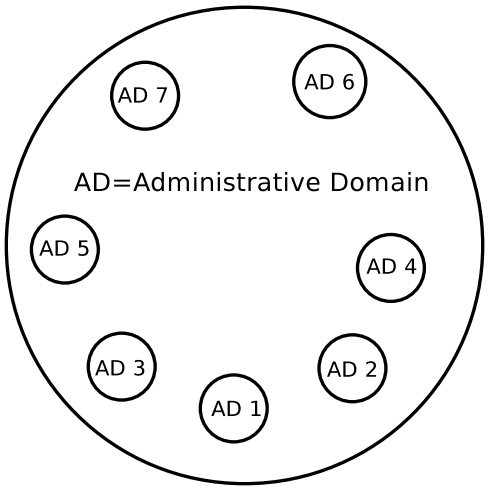


Figure 4.19: No Mechanism for Security Alert Sharing in Grid Computing Network Scenario

at the dashboard. The DSOC generates an alarm for almost every attempt. These alarms utilize network bandwidth in case of a multi-site network, and they use a high amount of disk space if the attacks continue for long periods of time. The reported security alarms are only available within the premises of one AD. This limitation does not suit grid networks as there are attacks that use computer worms which expands by themselves. These attacks could therefore expand to other members of a grid network. A mechanism has therefore been adopted in GSOC for sharing reported security alerts between multiple ADs to protect other members in a grid network for possible cross-domain attacks. See figure 4.19 where there is no security alert sharing mechanism. This scenario can lead to propagate some serious network attacks to other members of the grid. In figure 4.20 the security mechanism exists but it is not intelligent because ADs are sharing security alerts randomly. This sharing of information cannot be result oriented as it uses more network bandwidth and exposes internal security information to insecure ADs. Keeping these issues in mind, an efficient approach is to share the security alerts after classifying the ADs. See figure 3.13 for a depiction of security evaluation through the assignment of sharing mechanisms into three categories SL1 as most secure, SL2 as more secure, and SL3 as least secure. This classification gives a global view of security within a grid network to its members.

In figure 4.18 the GSOC detects the attack within one minute on the CBox. By the second minute they are detected on the ABox, which correlates the alerts coming from multiple local sites and discards false positives. By the third minute the details of the attacks are available on the SVOBox. By the fourth minute all alerts are ready for sharing with other ADs in a grid computing network. This can help blocking cross-

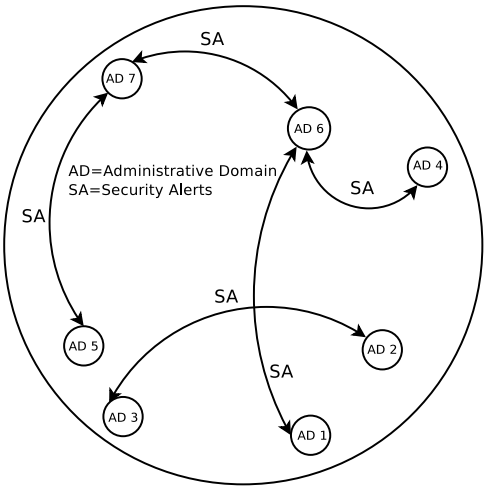


Figure 4.20: Unintelligent Mechanism for Security Alert Sharing in Grid Computing Network Scenario

domain attacks. It utilizes less bandwidth and disk space and gives a global view of security for the entire grid network.

4.4 Optimizing Detection of Distributed Attacks in Grid Computing Networks

To protect the grid computing network from attack propagation early detection of attack is very important. Early detection of malicious activities could lead towards the generation of false positives. These false positive make the security management systems unstable as discussed earlier. In order to minimize the false positives and accurately detect the attacks a distributed security alert summation technique has been adopted. It helps in detecting the attack by adding all the reported security alerts occurred during the same period of time within all the member of the grid computing network. Figures 4.21, 4.22 and 4.23 show early detection of SYN and Smurf attacks by summation of security alerts.

4.4.1 Smurf Attack detection

In this attack scenario the group of attackers launched the attack simultaneously on four ADs. They spoofed their IP addresses which makes it harder to detect early by the security management system. Using the summation of the sites the attacks are detected earlier. Multiple attackers use multiple commands of hping to launch the attacks, which are as under,

hping3 -a spoofed_IP_address -i u1 -S victim_IP_address

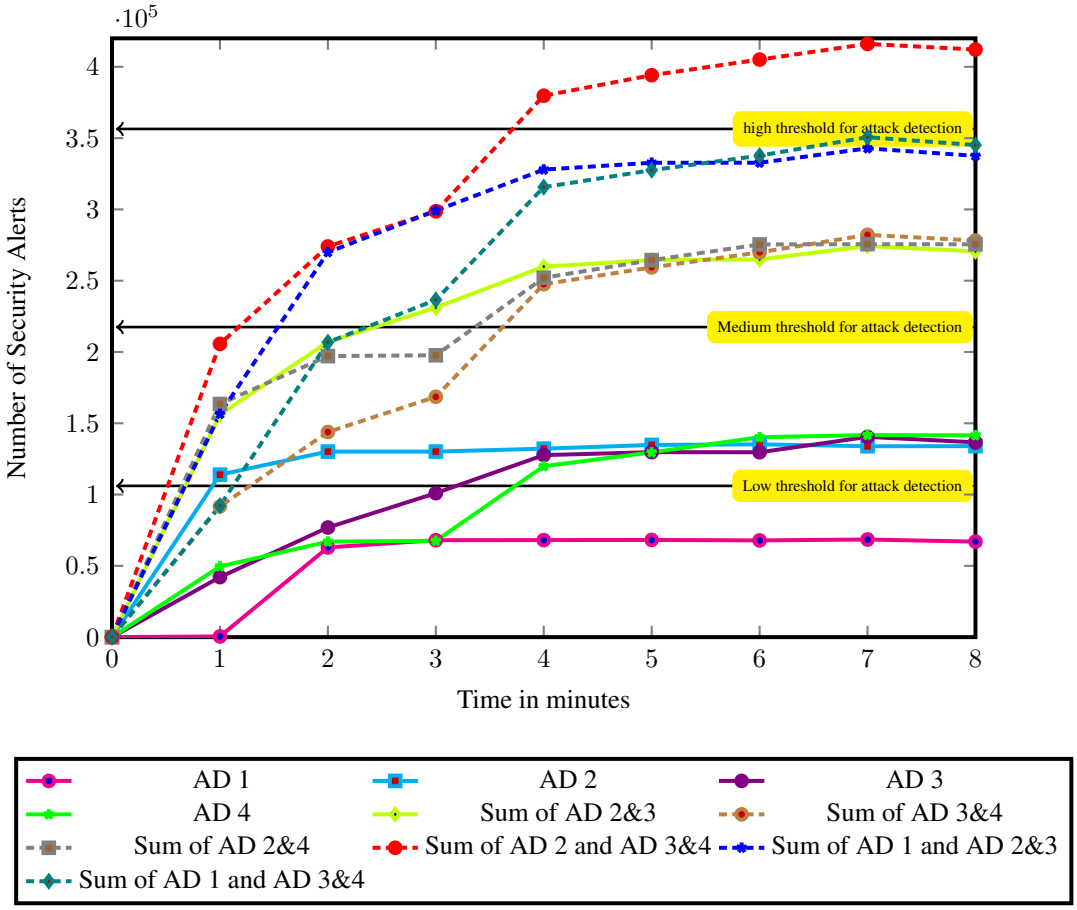


Figure 4.21: Detection of Smurf Attack

hping3 -a spoofed_IP_address -i u10 -S victim_IP_address
hping3 -a spoofed_IP_address -i u100 -S victim_IP_address
hping3 -a spoofed_IP_address -i u1000 -S victim_IP_address
The parameter of the hping command are, "-a" is used to set any IP address, "-i u" is the delay in transmitting another packet in micro seconds and "-S" is used to set the SYN TCP flag.

Figure 4.21 shows the early detection of Smurf attack using the security alerts summation mechanism. The received alerts from the members of the grid are further analyzed at each AD to detect distributed attacks. Three thresholds are set locally to detect the distributed attacks. The reason to set these thresholds is to detect the attacks as early as possible and to minimizing the false positives. The values are adjustable according to the size of the network and the capacity to handle and store the security alerts locally. In the experiments shown below I have given my values to these three

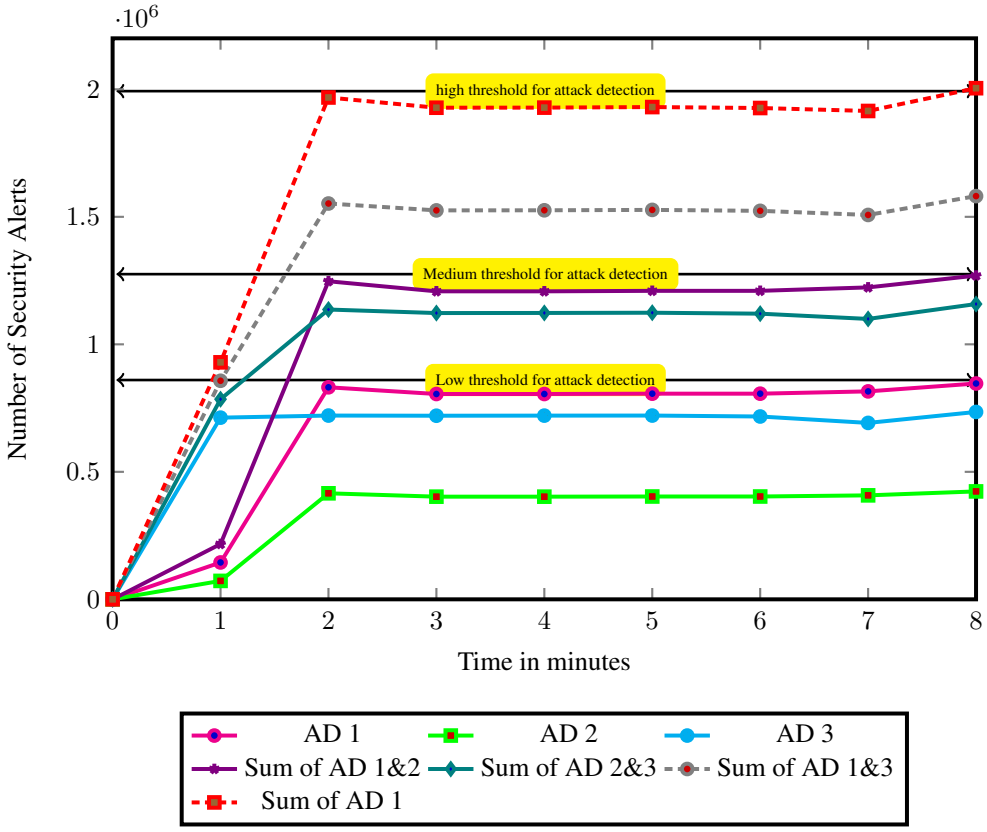


Figure 4.22: Detection of SYN Attack

levels. The low threshold is set to 100000 number of security alerts. The medium threshold is set to 250000 number of alerts. The high threshold is set to 400000 number of alerts. Using the low threshold the Smurf attack is detected on AD 1,3&4 between 1 to 2 minutes. The medium threshold detects the Sumurf attack on AD 3&4 is detectable between 3 to 4 minutes whereas on AD 2,3&4 the same attack is detected in between 1 to 2 minutes. The high threshold can only detect the summation of AD 2,3&4 at the fourth minute.

4.4.2 SYN Flooding Attack Detection

TCP protocol uses three-way handshake mechanism which is vulnerable to SYN flood attacks. (i) Attacker sends "SYN" to victim. (ii) Victim sends "SYN-ACK" back to the attacker. (iii) Attacker does not sends "ACK" and keeps sending "SYN" packet to the victim. The SYN flood attack is one kind of DoS attack where high number of "SYN" packets are send by the attackers to the victim machine. The victim machine allocates resources for each request sent by the attackers. The victim machine sends back the "SYN-ACK" to the source IP of the attackers. The attackers use the spoofed

IP addresses which does not exists in the network. This results in no "ACK" response from the attackers. The attackers continue sending high amount of "SYN" packets to the victim machine and victim waits for the "ACK" request. After some time these SYN packets sent by the attackers consume all the resources of the victim machine and makes it unstable which results in blocking all the requests coming from legitimate users. The SYN attack can halt the entire network operations if uses the broadcast address mixing with IP spoofing.

Figure 4.22 shows the SYN attack case where the thresholds are set to detect the instability in the security management system. Three ADs are shown in the graph which are the member of one grid network. The curves show that after passing of 2 minutes the ADs does not detect the attacks at the same rate and continue to work at a constant rate. This case occurs when the AD does not have much resources to handle the intense attacks. Here the summation of alert mechanism is very useful because it detects the attacks even when the security management systems are struggling. At the low threshold level the sum of AD 1&2 and the sum of AD 2&3 are detected in between 1&2 minutes. The medium alert threshold level detects the summation of alerts of the AD 1&3 and sum of AD 1,2&3 in between 1&2 minutes. The high alert threshold level only detects the summation of alerts of the AD 1,2&3 after 8 minutes. The hping command which is used to launch the attack by the attackers is given below,
hping3 –syn –destport 80 -i u1000 Victim_Machine
The parameter of the commands are, "–syn" is to set the SYN tcp flag, "–desport 80" is to set the port number of the victim and "-i u1000" is to send the packet after the delay of 1000 micro seconds. IPtable rules to detect the malicious packets is,
iptables -A INPUT -d 0/0 -s0/0 -p tcp –tcp-flags SYN,ACK,FIN,RST SYN -j LOG –log-prefix "SYN_ATTACK "
This rule logs the packets that are coming from any source to any destination using TCP protocol having any of the bit set SYN,ACK,FIN,RST. It saves every attempt of this type with the tag "SYN ATTACK" in the system logs.

4.4.3 Distributed SYN and PoD Attack Detection in Seconds

In this attack multiple attackers use SYN and PoD attacks together. Figure 4.23 shows the behavior of the SYN and PoD attack. The commands that are used for these attacks are,
hping3 –syn –destport 80 -i u10 Victim_Machine
hping3 –syn –destport 80 -i u100 Victim_Machine
ping -s 64000 -i 0 IP
ping -s 65000 -i 0 IP
The parameter of the Ping command are, "-s" defines the packet size, "-i" is the interval between two packets. Here "0" means it is set to flood mode.

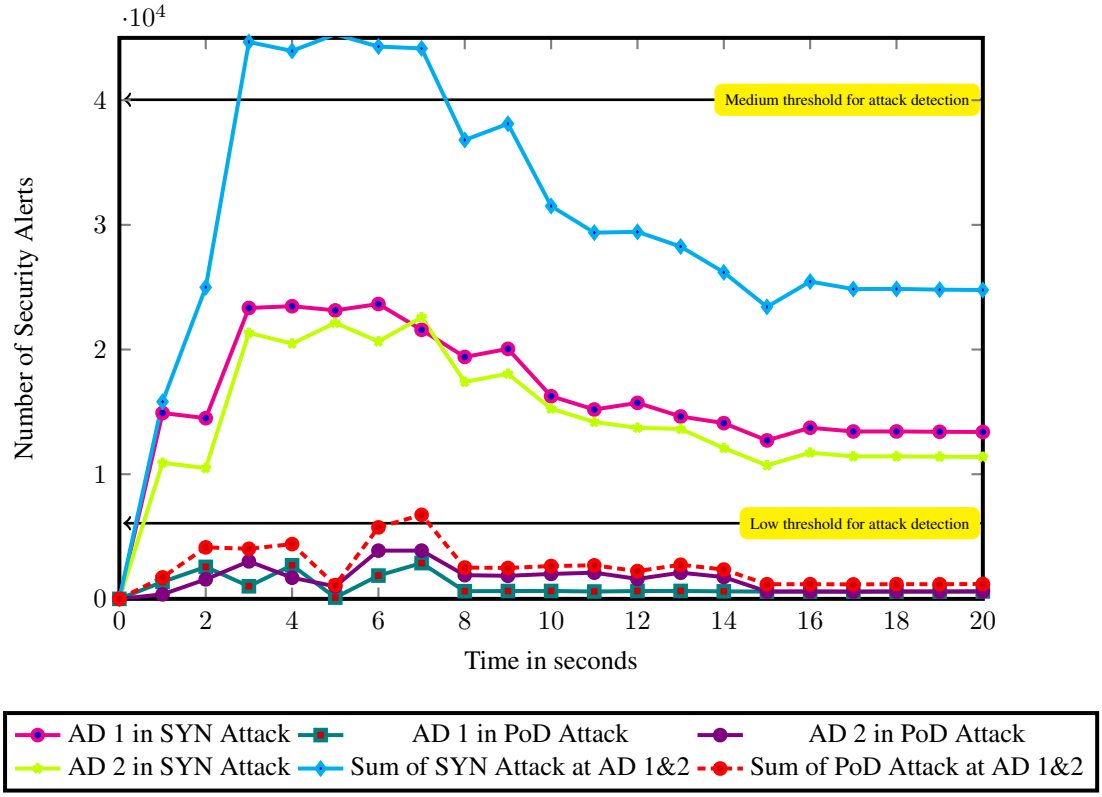


Figure 4.23: Multiple Attack Detection in Seconds

In this scenario the attackers increase and decrease their attack intensity in order to camouflage their malicious activities from the security management systems. Therefore the attack detection is more optimized by setting the thresholds in seconds. Multiple attackers launches the SYN and PoD attack at the AD 1 & 2. The objective of these attacks is to destabilize the security management system and hide the real attacks such as Brute Force. The graph shows that low threshold detects the distributed PoD attack on AD 1&2 in between 6 to 7 seconds. The medium threshold detects the distributed SYN attack on AD 1&2 in between 3 to 7 seconds.

5

Conclusion

THIS thesis covers the relevant research work that has been done in past for the security management and monitoring in grid computing networks. The classification of different security management systems has been shown in order to specify the area of security where GSOC has significant importance. Different parameters of these systems are compared and shown in the tabular form and their lacking to handle grid specific security issues.

5.1 Summary of GSOC

Each component of GSOC with its internal architecture and functionalities has been explained. The general view shows the placement of each component in the network. GSOC has a modular design which is scalable and handles grid specific properties. GSOC has a fault-tolerant capability that helps to continue detecting attacks even if its own components are under attack. The Basic Correlation (BC) and Advance correlation (AC) adoption in the design shows significant improvement in the performance and stability. The collected events are processed locally at each local site in order to extract the incidents of the attack. If some incidents are detected they are forwarded to the LA for further investigations. The further investigations include analyzing of reported alerts with the past and upcoming alerts in order to generate final alarm. The BC helps in minimizing network resources such as bandwidth, storage and processing power. The economized usage of network resources help to improve its stability and performance when the network is under intense distributed attacks. The AC helps to minimize the false positives because it takes 60 seconds to further analyze the alerts coming from multiple CBoxes and make sure that a final alarm contains the information of an actual attack and its source. The dynamic security evaluation based on mathematical equations that allow the members of the grid to be placed according

to the security levels. The static security evaluation is done by using OpenVAS, Saint and Nessus. The security alert sharing is a new concept applied in GSOC which helps restricting the propagations of the attacks and gives a global view of the security to the other members of the grid. The sharing of security alerts is fully customizable and only the alerts that are allowed are shared with other members of the grid.

The objective of the experiments is to show that GSOC is more stable compare to DSOC and OSSIM, the functionality of its components and how they behave to detect the attacks, its mechanism to block attack propagation and the minimization of false positive by summation of security alerts. Different tools are used to lunch multiple attacks simultaneously, to show that how GSOC processes, analyzes and reports distributed attacks. The core design of the GSOC with slight modification has a full capability to be extended and implemented in the cloud computing networks.

5.2 Future Study on GSOC

Deep Packet inspection (DPI) filters packets and their header for searching intrusions in the network. DPI is very effective in blocking spreading of worms in the network, virus infecting important files, and DoS attacks. It can detect IP Spoofing and obfuscation techniques used by the attackers. The future study to incorporate DPI in GSOC will enhance its attack detection capabilities.

Anomaly based intrusions are detected by misuse or abnormal behavior using heuristics rules. To add anomaly based attack detection in GSOC requires some time to train it by using artificial intelligence or neural networks. If anomaly based techniques will be applied in GSOC then the study on the false positive rates must be done with the current signature based technique because anomaly based systems generate high false positives.

GSOC is an attack monitoring and management systems for grid computing networks. The mechanisms of attack prevention are needed that will make it more effective to handle the security in grid computing networks. Some agents could be introduced and activated when required to achieve attack prevention at any site.

Security evaluation which is discussed in the chapter 3 is an area where exists difference of opinion among the security managers. A method can be proposed in GSOC where the security managers can exchange their security rules with each other. If all agreed the same security rules can be applied on all the members if the grid. The advantage of using the same security rules will be achieved in the form of homogeneous security evaluation of each member of the grid. The generation of low, medium and high security alerts will generate from the security rules used by the all the members of the grid. This proposition of security rules sharing can reduce the difference of

opinion among the security managers.

5.3 Future Work

Cloud computing provides Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). There exist multiple types of clouds such as community, private, public and hybrid clouds. Community clouds are formed by multiple organizations which are working for the same objective. Private cloud is owed by one administrative domain, this type of cloud network is used by organizations who wants to have full control of the cloud resources. Private cloud is not used by many users as the benefits are very limited. Public cloud are the most common and easily available place for user to use resources either free or pay as per usage. Hybrid clouds are the combination of community or public clouds. The member of these clouds can get benefits of both the cloud networks. It provides the users a large range of resources available to use. There exists many cloud service provider and they are growing as the technology is becoming mature. The most prominent are the Amazon Elastic Compute Cloud (Amazon EC2) [91] or S3 [92], Google cloud services [93], Eucalyptus [94], IBM smart cloud [95] and Opennebula [96]. GSOC could be deployed in any type but its significance is more if deployed in hybrid and public clouds with some modifications. GSOC can give the global view of the security of the inter and intra cloud infrastructures. The security events and alerts can be shared and correlated to detect attacks originated from different attacker using multiple cloud networks. Similarly security evaluation can be performed using GSOC to identify vulnerabilities present in the cloud networks. Figure 5.1 shows the internal architecture of the security management system for cloud networks. The main controller of the systems is the manager that holds the security events coming from different cloud networks. It will format the received events and correlate them to report the security alerts. The reported security alerts are further analyzed with the general security rules, if matched with any, an alarm will generate containing all the necessary information about the attack.

The most common security concern in the cloud computing which is raised by the experts, is the handling of the data that is going to be placed at the service provider's network. The data placed on the cloud can be misused or compromised and the owner does not even now about the incident. GSOC separate policies can be applied on the data in the form of security rules. The security rules generate alerts if the security policy is violated by the cloud service provider, by the owner itself or by any of its staff members. This solution can also helps in resolving issue of privacy in the cloud infrastructure.

To deploy GSOC in cloud computing infrastructure certain modifications needs to be done, some of the propositions are:

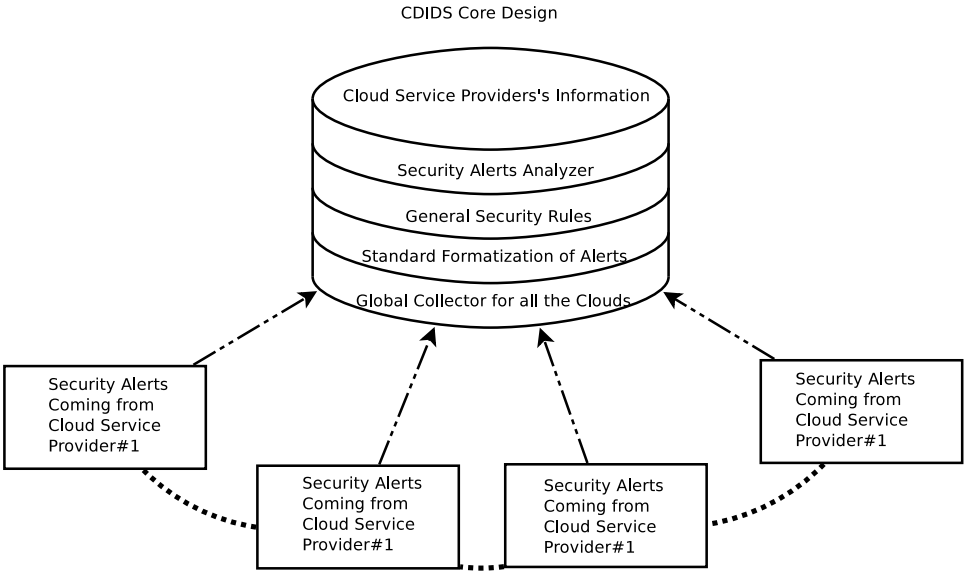


Figure 5.1: Cloud Distributed Intrusion Detection (CDIDS) Core Design

(i) New boxes should be introduced at each service, that means one box for infrastructure, one for platform and one for software.

(ii) These boxes must be programmed to handle the specific security issues occurred at each service level.

(iii) These boxes must works separately and report to the LA, but if needed can also collaborate to detect the attacks which are launched using all the three service levels.

(iv) For every service there can be a separate LA.

(v) All the LAs at each service level must report to the GA which will handle the security of the entire cloud.

(vi) There may exist multiple SVOBoxes.

(vii) The manager of all the boxes will be the Cloud Box.

Figure 5.2 represents the overview of the cloud security management system. It shows that how events are collected and correlated to detect the attacks occurring in different cloud networks. The cloud security management system helps to handle under given types of issues:
(i) Early detection of Attacks by summation of alerts locally and globally. Summation of alerts can be done securely by using Secure Multi-party Computation (SMC). The SMC is first proposed by Andrew Chi in [97].

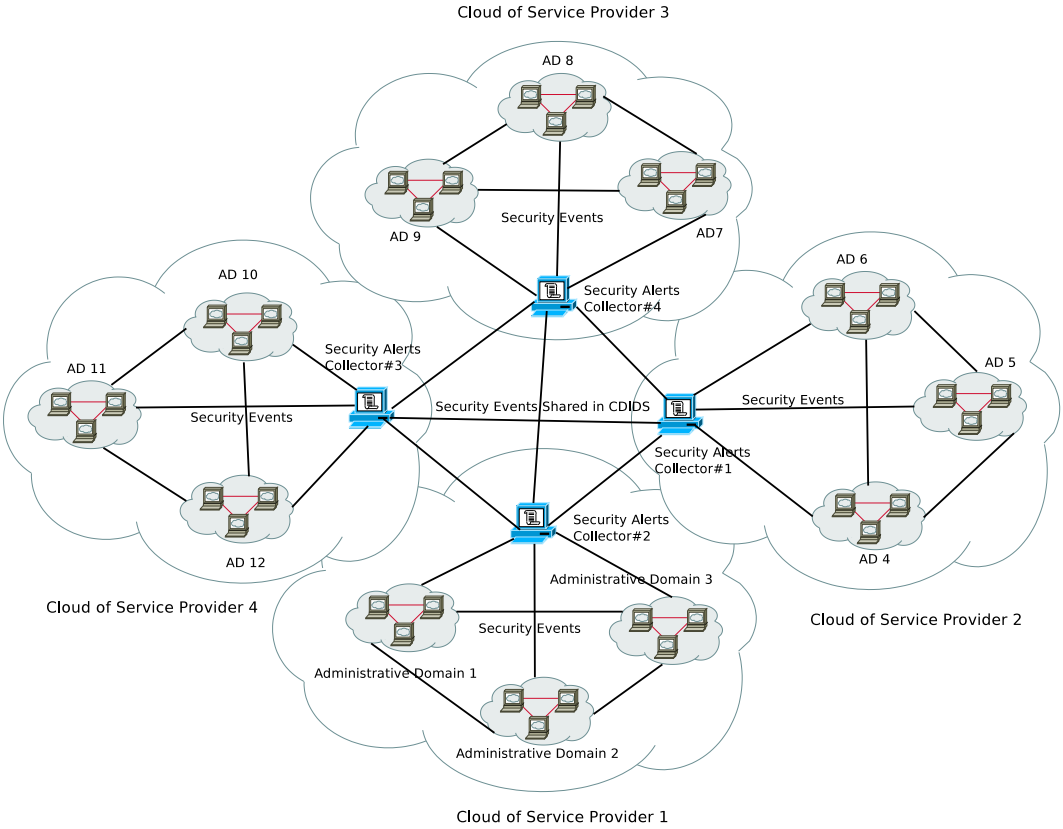


Figure 5.2: Intra and Inter-public Cloud Architecture

(ii) Detection of very powerful attacks which last for very short period of time.
(iii) Fault Tolerance in case if one of the collecting box fails the other one takes it place and report to the manager for that incident.
(iv) Correlation and analysis of security events to help in minimization of false positives.
(v) Security alerts sharing mechanism between intra cloud can be further improved using Kth Anonymity discussed in [98] and [99].

The most recent issues are discussed by Balduzzi et al. [100] where they highlighted the security issues present in the public virtual images. They performed vulnerability tests on 5000 virtual machine images available in four different data centers of Amazon [91] and reported several security issues, some of them are: (i) The confidential files were deleted while preparing the virtual machine image but these file are easily recoverable such as password files, SSH private keys, PGP private keys, etc. (ii) Discovered instances of SSH, different services and Web. (iii) History of files of VNC, MySql, DNS, WebApp, and SQL. Bugiel et al. [101] also highlighted the similar issues discussed by Balduzzi et al. [100], but they performed experiments on 1255 Amazon images and the scope of their experiments was limited in covering security issues. The main focus of both the findings is to emphasis that there exist some serious security threats in cloud computing infrastructures. Garfinkel and Rosenblum [102] highlighted the use of third party virtual images and their security issues. They also discussed other security issues exist in user generated virtual images. Glott et al. [103] highlighted the security issues that occurs when the virtual images are shared within multiple users in cloud infrastructure. They proposed some assessments to find out the vulnerabilities present in the virtual image. Ristenpart et al. [104] only presented the introduction of side channel attacks in cloud computing networks. Bleikertz at al. [105] used graph theory techniques to deploy virtual machine images in AmazonEC2 infrastructure. The objective of their work is to focus on the security issues that are present at the infrastructure level which is a different approach as others focus more on the virtual images security. Their propositions are based on configuring network and setting the security policies properly.

Bibliography

[1] Open Source Security Information and Event Management (OSSIM). Access from:http://alienvault.com/resources/documentation/technical-documentation, 2011.

[2] Prelude is a universal "Security Information Event Management" (SIEM) system. Available at: www.prelude-technologies.com/en/development/documentation/index.html, 2012.

[3] Akab is a modular and scalable SIEM+ (Security Information Event Management). Available at: www.araknos.it/en/prodotti/architettura-akab.html, 2012.

[4] Stuart Kenny and Brian Coghlan. Towards a grid-wide intrusion detection system. In *Advances in Grid Computing - EGC 2005*, volume 3470 of *Lecture Notes in Computer Science*, pages 275–284. Springer Berlin / Heidelberg, 2005.

[5] Fang Yie Leu, Jia Chun Lin, Ming Chang Li, Chao Tung Yang, and Po Chi Shih. Integrating grid with intrusion detection. *International Conference on Advanced Information Networking and Applications, Volume*, 1:304–309, 2005.

[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28:18 – 28, 2009.

[7] Renaud Bidou, Julien Bourgeois, and François Spies. Towards a global security architecture for intrusion detection and reaction management. In K. Chae and M. Yung, editors, *Proc. of the 4th Int. Ws. on Information Security Applications, WISA 2003*, volume 2908 of *LNCS*, pages 129–142, Jeju, Corea, August 2003.

[8] Abdoul Karim Ganame, Julien Bourgeois, Renaud Bidou, and François Spies. A high performance system for intrusion detection and reaction management. *Journal of Information Assurance and Security*, 3:181–194, September 2006.

[9] Open Source Vulnerabilities Data Base (OSVDB). Available at: http://osvdb.org/.

[10] Christine Morin. Executive summary-3:building and promoting a linux-based operating system to support virtual organizations for next generation grids. Available at: https://www.xtreemos.org/project/project-status, May 2009.

[11] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Number 978-1558604759. Morgan Kaufmann, 1 edition, August 1998.

[12] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid - enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15:2001, 2001.

[13] K. Krauter, R. Buyya, and M. Maheswaran. A taxonomy and survey of grid resource management systems for distributed computing. *Software: Practice and Experience*, 32(2):135–164, 2002.

[14] Franco Travostino, Joe Mambretti, and Gigi Karmous-Edwards, editors. *Grid Networks: Enabling Grids with Advanced Communication Technology*. Number 978-0470017487. Wiley, September 2006.

[15] Frederic Magoules, Thi-Mai-Huong Nguyen, and Lei Yu. *Grid Resource Management: Toward Virtual and Services Compliant Grid Computing*. Number 978-1420074048. CRC Press, 1 edition, September 2008.

[16] Lightweight Middleware for Grid Computing (gLite). Available at: http://glite.cern.ch/, 2010.

[17] UNICORE (Uniform Interface to Computing Resources). Available at: `http://www.unicore.eu/UNICORE`.

[18] Grid-enabled know-how sharing technology based on arc services and open standards (knowarc). Available at: http://www.knowarc.eu/.

[19] Dawid Kurzyniec, Magdalena Slawinska, Jaroslaw Slawinski, and Vaidy Sunderam. Unibus: a contrarian approach to grid computing. *J. Supercomput.*, 42(1):125–144, October 2007.

[20] Enabling Grids for E-sciencE (EGEE). Available at: http://www.eu-egee.org/, 2010.

[21] NGI-DE is the National Grid Initiative for Germany. Available at : www.ngi-de.eu/english/index.php.

[22] BiG is the grid infrastructures for scientific research in the netherlands. Available at : http://www.biggrid.nl/.

[23] The National Grid Service (NGS) to facilitate uk research. Available at : http://www.ngs.ac.uk/.

[24] BEgrid is the computing/data grid infrastructure of the belgian grid for research. Available at : http://www.begrid.be/.

[25] AstroGrid is the doorway to the Virtual Observatory (VO). Available at : http://www.astrogrid.org/.

[26] The national cancer institute launched the Cancer Biomedical Informatics Grid (CaBIG). Available at : https://cabig.nci.nih.gov/.

[27] National center for atmospheric research. Available at : http://www.earthsystemgrid.org/home.htm.

[28] Economic paradigm for "resource management and scheduling" for service-oriented grid computing. Available at : http://www.buyya.com/ecogrid/.

[29] Network for earthquake engineering simulation. Available at : http://nees.org/.

[30] The national fusion collaboratory project. Available at : http://www.fusiongrid.org/projects/.

[31] Biomedical informatics research network. Available at : http://www.birncommunity.org/.

[32] Grid-based e-infrastructure for data archiving/communication and computationally intensive applications in the medical sciences. Available at : http://www.neugrid.eu/pagine/home.php.

[33] SETI@home is a scientific experiment that uses internet-connected computers in the search for extraterrestrial intelligence (SETI). Available at : http://setiathome.ssl.berkeley.edu/.

[34] Enterthegrid is the largest directory on grid computing. Available at : http://www.enterthegrid.com/.

[35] Maozhen Li and Mark Baker. *The Grid: Core Technologies*. Number 978-0470094174. Wiley, 1 edition, May 2005.

[36] Snort network intrusion prevention and detection system (IDS/IPS). Available from: http://www.snort.org/, 2012.

[37] Fang Yie Leu, Jia Chun Lin, Ming Chang Li, and Chao Tung Yang. A performance-based grid intrusion detection system. In *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05) Volume 1*, pages 525–530, Washington, DC, USA, 2005. IEEE Computer Society.

[38] Fang Yie Leu, Ming Chang Li, and Jia Chun Lin. Intrusion detection based on grid. In *Proceedings of the International Multi-Conference on Computing in the Global Information Technology*, pages 62–67, Washington, DC, USA, 2006. IEEE Computer Society.

[39] Yonggang Chu, Jun Li, and Yixian Yang. The architecture of the large-scale distributed intrusion detection system. In *PDCAT'05*, pages 130–133, 2005.

[40] Paulo F. Silva, Carlos B. Westphall, Carla M. Westphall, and Marcos D. Assunção. Composition of a dids by integrating heterogeneous idss on grids. In *Proceedings of the 4th international workshop on Middleware for grid computing*, MCG '06. ACM, 2006.

[41] Yang Xiang and Wanlei Zhou. Protect grids from DDoS attacks. In *GCC*, pages 309–316, 2004.

[42] Ming Li, Chi-Hung Chi, Weijia Jia, Wei Zhao, Wanlei Zhou, Jiannong Cao, Dongyang Long, and Qiang Meng. Decision analysis of statistically detecting distributed denial-of-service flooding attacks. *International Journal of Information Technology and Decision Making*, 2(3):397–405, 2003.

[43] Scalable Simulation Framework (SSF): A public-domain standard for discrete-event simulation of large, complex systems in Java and C++.

[44] Ong Tian Choon and A. Samsudin. Grid-based intrusion detection system. *Communications, 2003. APCC 2003. The 9th Asia-Pacific Conference on*, 3(0-7803-8114-9):1028– 1032, 21-24 Sept 2003.

[45] Globus: Grid Security Infrastructure (GSI). http://www-unix.globus.org/toolkit/docs/3.2/security.html, 2010.

[46] Von Welch, Jarek Gawor, Carl Kesselman, Sam Meder, and Laura Pearlman. Security for grid services. In *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, pages 48–57. IEEE Press, 2003.

[47] Ian Foster, Carl Kesselman, Jeffrey M.Nick, and Steven Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. Available at: http://www.globus.org/alliance/publications/papers/ogsa.pdf, 2002.

[48] The open grid services architecture, version 1.5. Available at: http://www.ogf.org/documents/GFD.80.pdf, 2002-2006.

[49] Richard Ford, Mark Bush, and Alexander Bulatov. Predation and the cost of replication: New approaches to malware prevention? *Computers & Security*, 25(4):257–264, 2006.

[50] Helen J. Wang, Chuanxiong Guo, Daniel R. Simon, and Alf Zugenmaier. Shield: vulnerability-driven network filters for preventing known vulnerability exploits. *SIGCOMM Comput. Commun. Rev.*, 34:193–204, August 2004.

[51] Phil Porras, Dan Schnackenberg, Stuart Staniford-Chen, Maureen Stillman, and Felix Wu. The Common Intrusion Detection Framework Architecture (CIDF). Available at: http://gost.isi.edu/cidf/drafts/architecture.txt, 1998.

[52] Stuart Staniford-Chen, Brian Tung, Phil Porras, Cliff Kahn, Dan Schnackenberg, Rich Feiertag, and Maureen Stillman. The common intrusion detection framework - data formats. http://tools.ietf.org/html/draft-staniford-cidf-data-formats-00, September 1998.

[53] CIDF Working Group. The Commun Intrusion Specification Language (CISL), http://gost.isi.edu/cidf/drafts/language.txt, 1999.

[54] B. Tung. The common intrusion specification language: A retrospective. In *In Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, January 2000.

[55] Abdoul Karim Ganame, Julien Bourgeois, Renaud Bidou, and François Spies. A global security architecture for intrusion detection on computer networks. *Computers & Security*, 27(1-2):30–47, 2008.

[56] Wael Kanoun, Nora Cuppens-Boulahia, Frédéric Cuppens, Samuel Dubus, and Antony Martin. Success likelihood of ongoing attacks for intrusion detection and response systems. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, pages 83–91, Washington, DC, USA, 2009. IEEE Computer Society.

[57] Reza Sadoddin and Ali Ghorbani. Alert correlation survey: framework and techniques. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, PST '06, pages 1–10. ACM, 2006.

[58] William R. Cheswick, Steven M. Bellovin, and Aviel D. Rubin. *Firewalls and Internet Security; Repelling the Wily Hacker*. Addison-Wesley, Reading, MA, second edition, 2003.

[59] Van Hauser. The hacker's choice, a very fast network logon cracker which support many different services. Available at: http://freeworld.thc.org/, 2010.

[60] Guess who is a password brute force utility for attacking secure shell version 2 accounts. Available at: http://www.vulnerabilityassessment.co.uk/guesswho.htm, 2010.

[61] Bart Jacob, Michael Brown, Kentaro Fukui, and Nihar Trivedi. *Introduction to Grid Computing*. IBM Corp., December 2005.

[62] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820 (Proposed Standard), June 2004.

[63] Marco Cremonini, Sabrina De Capitani di Vimercati, Ernesto Damiani, and Pierangela Samarati. An xml-based approach to combine firewalls and web services security specifications. In *Proceedings of the 2003 ACM workshop on XML security*, XMLSEC '03, pages 69–78. ACM, 2003.

[64] Anirban Chakrabarti. *Grid Computing Security*. Springer, 2007.

[65] Public-Key Infrastructure (PKI). Available at: http://datatracker.ietf.org/wg/pkix/charter/, 2011.

[66] Erin Cody, Raj Sharman, H. Raghav Rao, and Shambhu J. Upadhyaya. Security in grid computing: A review and synthesis. *Decision Support Systems*, 44(4):749–764, 2008.

[67] Uwe Schwiegelshohn, Rosa M. Badia, Marian Bubak, Marco Danelutto, Schahram Dustdar, Fabrizio Gagliardi, Alfred Geiger, Ladislav Hluchý, Dieter Kranzlmüller, Erwin Laure, Thierry Priol, Alexander Reinefeld, Michael M. Resch, Andreas Reuter, Otto Rienhoff, Thomas Rüter, Peter M. A. Sloot, Domenico Talia, Klaus Ullmann, and Ramin Yahyapour. Perspectives on grid computing. *Future Generation Comp. Syst.*, 26(8):1104–1115, 2010.

[68] Massimo Coppola, Yvon Jégou, Brian Matthews, Christine Morin, Luis Pablo Prieto, Oscar David Sánchez, Erica Y. Yang, and Haiyan Yu. Virtual organization support within a grid-wide operating system. *IEEE Internet Computing*, 12(2):20–28, 2008.

[69] Julien Bourgeois and Syed Raheel Hassan. Managing security of grid architecture with a grid security operation center. In *SECRYPT'09, Int. Conf. on Security and Cryptography, Milan, Italy*, pages 403–408. INSTICC Press, July 2009.

[70] Syed Raheel Hassan, Jasmina Pazardzievska, and Julien Bourgeois. Minimization of security alerts under denial of service attacks in grid computing networks. In *The Int'l Conf. Grid Computing and Applications (GCA11)*, pages 44–50, Las Vegas, USA, July 2011. WORLDCOMP11, CSREA Press.

[71] Common vulnerabilities and exposures is a dictionary of publicly known information security vulnerabilities and exposures. Available at: http://cve.mitre.org/, 2010.

[72] J. Ganame, A.K. Bourgeois. Defining a simple metric for real-time security level evaluation of multi-sites networks. volume 14-18, pages 1 – 8, April 2008.

[73] Christopher Kruegel. *Intrusion Detection and Correlation: Challenges and Solutions*. Springer-Verlag TELOS, Santa Clara, CA, USA, 2004.

[74] Christopher Kruegel, Fredrik Valeur, Giovanni Vigna, and Richard Kemmerer. Stateful intrusion detection for high-speed networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 285–294. IEEE Press, 2002.

[75] Richard A.Kemmerer and Giovami Vigna. Intrusion detection: A brief history and overview. *IEEE Computer*, pages 27–30, 2002.

[76] Security scanner for oracle and various flavors of unix. Available at: www.nessus.org/, 2010.

[77] The Open Vulnerability Assessment System (OpenVAS). Available at: www.openvas.org/, 2010.

[78] Vulnerability Management, Assessment, Penetration Testing (SAINT). Available at: www.saintcorporation.com/, 2010.

[79] RSnake and John Kinsella. Slowloris HTTP Denail of Service. Available at: http://ha.ckers.org/slowloris/, 2010.

[80] Hping is a command-line oriented TCP/IP packet assembler/analyzer. Available at : http://www.hping.org/, 2011.

[81] XtreemOS is a grid operating system based on linux. Available at : http://xtreemos.org/, 2012.

[82] Christine Morin. Executive summary-1:building and promoting a linux-based operating system to support virtual organizations for next generation grids. Available at: https://www.xtreemos.org/project/project-status, May 2007.

[83] Kerrighed is a Single System Image operating system for clusters. Available at : http://www.kerrighed.org/wiki/index.php/Main_Page, 2012.

[84] Christine Morin. Executive summary-2:building and promoting a linux-based operating system to support virtual organizations for next generation grids. Available at: https://www.xtreemos.org/project/project-status, May 2008.

[85] Rémy Garrigue and Yann Radenac. XtreemOS Guide. Available at : http://xtreemos.org/documentation-support, March 2012.

[86] M. Wilson and Alvaro Arenas. State-of-the-Art in trust & security for OS and Grids. Available at : http://xtreemos.org/project/publications, June 2006.

[87] Grid'5000 is a scientific instrument for the study of large scale parallel and distributed systems. Access from: https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home, 2012.

[88] French National Telecommunication Network for Technology, Education and Research (RENATER). Access from : http://www.renater.fr/?lang=en, 2012.

[89] Sventek. Apsend is a TCP/IP packet sender to test firewalls and other network applications. Available at: http://packetstormsecurity.org/, 2010.

[90] JoMo-Kun. Medusa is intended to be a speedy, massively parallel, modular, login brute-forcer, 2010.

[91] Amazon Elastic Compute Cloud (Amazon EC2). Available at : http://aws.amazon.com/ec2/, 2012.

[92] Amazon simple storage service (amazon s3). Available at : http://aws.amazon.com/s3/, 2012.

[93] Google?s cloud services. Available at : http://www.google.com/enterprise/cloud/, 2012.

[94] Eucalyptus is the world's most widely deployed cloud computing software platform for on-premise (private) infrastructure as a service clouds. Available at : http://www.eucalyptus.com/, 2012.

[95] Smartcloud is the ibm vision for cloud computing. Available at : http://www.ibm.com/cloud-computing/us/en/, 2012.

[96] Opennebula is an open-source project. Available at : http://opennebula.org/cloud:usingit, 2012.

[97] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164, 1982.

[98] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer-Verlag, 2007.

[99] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[100] Marco Balduzzi, Jonas Zaddach, Davide Balzarotti, Engin Kirda, and Sergio Loureiro. A security analysis of amazon's elastic compute cloud service. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1427–1434. ACM, 2012.

[101] Sven Bugiel, Stefan Nürnberger, Thomas Pöppelmann, Ahmad-Reza Sadeghi, and Thomas Schneider. Amazonia: when elasticity snaps back. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 389–400. ACM, 2011.

[102] Tal Garfinkel and Mendel Rosenblum. When virtual is harder than real: security challenges in virtual machine based computing environments. In *Proceedings of the 10th conference on Hot Topics in Operating Systems - Volume 10*, HOTOS'05, Berkeley, CA, USA, 2005. USENIX Association.

[103] Rdiger Glott, Elmar Husmann, Ahmad-Reza Sadeghi, and Matthias Schunter. Trustworthy clouds underpinning the future internet. In *IEEE Signal Process. Lett.'11*, pages 209–222, 2011.

[104] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 199–212. ACM, 2009.

[105] Sören Bleikertz, Matthias Schunter, Christian W. Probst, Dimitrios Pendarakis, and Konrad Eriksson. Security audits of multi-tier virtual infrastructures in public infrastructure clouds. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, CCSW '10, pages 93–102. ACM, 2010.