Minimization of Security Alerts under Denial of Service Attacks in Grid Computing Networks

Syed Raheel Hassan[1], **Jasmina Pazardzievska**[2], **and Julien Bourgeois**[1]
[1]Laboratory of Computer Science, University of Franche-Comte (UFC), Leprince-Ringuet, Montbeliard, France
[2]Faculty of Electrical Engineering. and IT, University Ss. Cyril and Methodius, Skopje, Republic of Macedonia

Abstract— *Grid computing networks aggregate huge computing power that 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 security management solutions to sabotage the normal operations of the grid computing network. In this paper the design of the grid SOC (GSOC) which minimizes the huge security alerts generated under network attacks will be discussed. GSOC performance has been compared with the DSOC and its attack detection capabilities with Snort and some experiments are presented using Grid'5000 network.*

Keywords: GSOC, DoS in Grid Computing Networks, Minimization of Security Alerts using GSOC.

1. Introduction

In recent years, different multi-administrative domains started working together as one grid network. The emergence of different organizations has made the grid computing network vulnerable to many network attacks. Due to the nature of the grid an attacker can use the grid computational power to target any administrative domain attached to the grid network for example Distributed Denial of Service (DDoS) attacks. When an attacker launches an intensive DDoS attack on the network the IDS starts generating many security alerts. It starts sending these alerts to the central database. This huge number of security alerts can create bottlenecks in the network and uses lots of disk space. Due to these intensive attacks the IDS can becomes so overloaded and therefore turning unstable. This instability results in the creation of many security alerts or some-times in false positives. Both the instability and the huge number of security alerts that administrator has to manage, give the attacker a fair chance to perform malicious activities. The instability of an IDS is due to multiple reasons. The most common ones observed are due to disk space failure, database failure and system process queue overloading. Intrusion detection and prevention systems (IDPSs) have been introduced to help the network administrator thwart possible network attacks. At present, IDPSs are also struggling to efficiently protect multi-administrative domain networks which can change their size dynamically. They hardly achieve this goal and reducing number of false positives while maintaining performance is still an issue [1]. The remaining parts of the paper are organized as section 2 presents the related work, in section 3 the architecture of GSOC has been discussed along with its components. Experiments are to be found in section 4 and the conclusion in section 5 .

2. Related Work

Protect grids from DDoS Attacks by Yang Xiang and Wanlei Zhou [2] proposed a distributed defense system for detecting DDoS attacks. This system requires access to the routers of each site. They performed the tests on the SSFNet (Scalable Simulation Framework) [3] which allowed them to capture and analyze all the network traffic between different sites. Their solution lacks practical implications because the access to routers and the capturing of network traffic of external sites is not possible in real active grid networks.

Security for Grid Service by Von Welch et al. [4] was the work done to upgrade the Globus Toolkit version 2 (GT2) so that it became the Globus Toolkit version 3 (GT3). It was the first implementation of the Open Grid Service Architecture (OGSA). OGSA was first suggested by Ian Foster in [5]. The OGSA provides heterogeneous systems with interoperability in order to communicate with different types of resources. The technical documentation of the OGSA 1.5 version which is available at [6] recommends to use intrusion detection systems for handling DDoS attacks on grid services. The OGSA does not provide any mechanism with how to counter DDoS attacks from the trusted user.

Predation and the cost of replication: New approaches to malware prevention by Richard Ford et al. [7] have used a ++shield program which was a modified version of the shield program. The Shield was developed by Wang et al. [8] It limits Malicious Mobile Code (MMC) in the network. In their experiments of shield heuristic simulation they have used the improved version of shield that was installed by default in all machines. If any machine has been attacked, the victim machine blocks the attack attempts by returning a magic number into the TCP headers or the

packet payload. This technique was useful to overcome DoS attacks but could not handle DDoS attacks. The DDoS uses multiple sources such as the attacker with mock IP addresses. Therefore even if the attacked machine keeps blocking the requests, it cannot handle DDoS attacks.

Distributed Security Operation Center (DSOC) was proposed by Ganame et al. [9]. It shows better stability in multi-site networks by detecting DDoS attacks. When the DSOC was deployed in the grid computing networks, it did not give consistent results. It does not handle the grid specific properties namely, (i) The grid network, a combination of different administrative domains, each of them composed of multi-site networks. (ii) In grid network a high number of nodes collaborate with one another. Therefore the size of the network is increasing and decreasing dynamically. (iii) In grid network a view of the security events of external networks is unavailable. (iv) The DSOC under DDoS attacks in a grid network needs much more disk space as it does not have time-based correlation modules.

Keeping the above-mentioned issues in view, the GSOC has been proposed by Bourgeois and Hassan [10]. It overcomes the limitation of the DSOC. The GSOC has two levels of correlation namely, basic and advance which help the GSOC to detect more sophisticated and distributed attacks. Due to this two-step correlation, the GSOC reduces the size of logs at both the collector and the analyzing ends. The aim of our work is to develop a security operation center dedicated to multi-administrative domain networks.

3. Grid Security Operation Center (GSOC) Architecture

In this section the components of the GSOC are explained with the importance of correlation in detecting complex attacks. The GSOC is based on the concept of separate boxes [11] that perform a specific task. The GSOC has four main components which are an event-generating box (EBox), a collecting box (CBox), a Local Analyzer (LA) which consists of a database box (DBox) and an alert-analyzing box (ABox) and a Global Analyzer which contains a global intrusion data base called (gidb).

3.1 Correlation

The main purpose of correlation is to analyze complex information sequences and to produce simple, synthesized, real-time alerts. The GSOC introduces two-level correlations: (i) Basic Correlation (BC) and (ii) Advanced Correlation (AC). This two-level hierarchy reduces the network traffic between the GSOC components and causes an easier detection of complex intrusions. The CBox has the role of performing basic correlation, whereas the LA is responsible for advanced correlation. The main purpose of BC is to reduce the network load between the GSOC modules; therefore attack detection is easier to perform. BC does not have the ability of detecting 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.

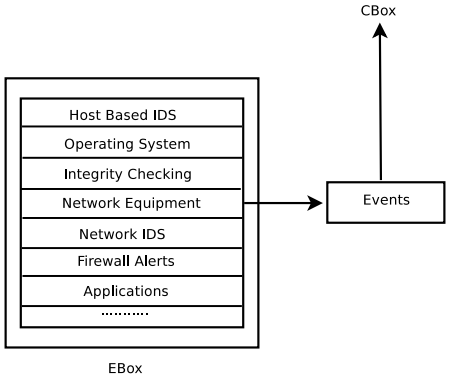

Fig. 1: EBox Design

3.2 Event-Generating Box (EBox)

The EBox is a component in grid network that generates events (see figure 1). These events could be of two types. One from the sensors which generates data due to any operation performed on them, this includes operating systems, firewalls, routers, switches, wireless HUBs or RADIUS servers. The second type 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 ping or snmp [12]. These events are then forwarded to the CBox.

3.3 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 are too high. Every EBox has a different format for reporting the event. Therefore the CBox collects this raw information from different protocols shown in figure 3. The dispatcher that plays an intermediary role is placed between the event-receiving protocols and the application modules. The application modules are the modules in the CBox which contains the possible attack lists. The dispatcher searches for these reported events from the EBox and tries to match them within available application modules like Linux, Windows and XtreemOS. When the reported event matches

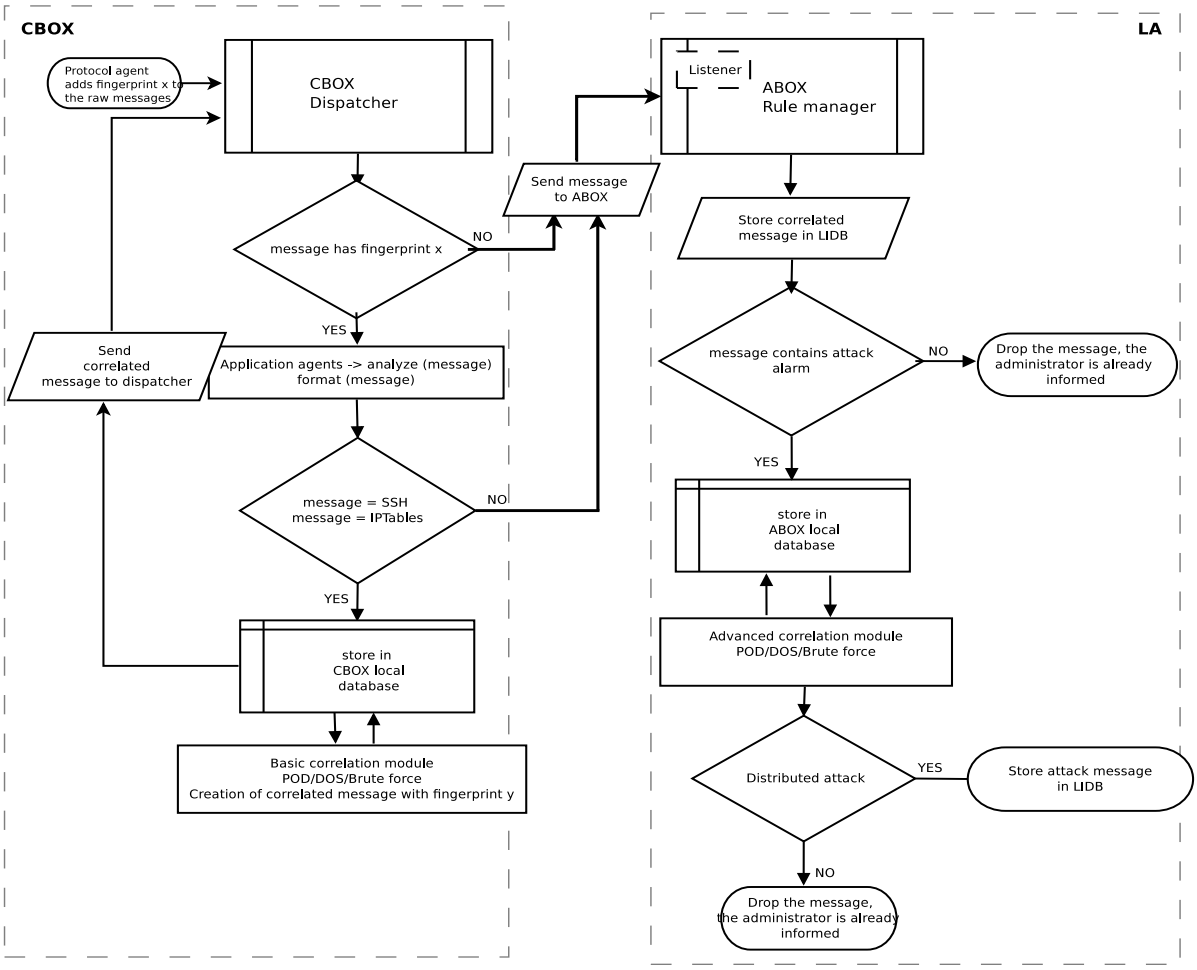

Fig. 2: Basic and Advanced Correlation flow chart

any defined attack template, it is then arranged in an internal format before it is sent to Basic Correlation.

3.3.1 Basic Correlation

The left part of figure 2 is the detailed explanation of basic correlation. The basic correlation module can be thought as a message marker. Each message is labeled depending on its contents, it checks if a message is containing an attack alert or it is a regular message. Each raw message sent from the EBoxes and received by the event-receiving protocols at the CBox is labeled with fingerprint (eg: fingerprint x). This fingerprint points out to the dispatcher that this message should be first analyzed by the application modules and if supported rule is found then it will be formatted. The dispatcher inspects whether these formatted messages are the ones that the administrator is interested in correlating (message originating from ssh session or message from IPtable rules at the EBoxes). If this condition is true, these kinds of messages are stored in a local database for a very short period of time (at most one minute). If this condition is not true, the CBox forwards that message to the LA (specifically ABox) in order to display a global view of the whole network. After the basic correlation new fingerprint (eg: fingerprint y), different than the one added to the raw messages is applied to the stored messages in the local database. This fingerprint tells the dispatcher that this message has already correlated and should be sent to the LA for further analysis. Afterwards, only the correlated messages are stored in the local database and transferred to the dispatcher which further forwards them to the LA. At this stage the messages that contain an attack are forwarded to the LA as well as those that are not containing any attacks. The communication between the CBox and the LA is over socket protocol.

3.4 Local Analyzer (LA)

The Local Analyzer is composed of two modules (i)alert analyzing box (ABox) and (ii)database box (DBox) (see figure 4). The *ABox* job is to report the alerts of the messages received from the CBox. All the CBoxes from the multiple

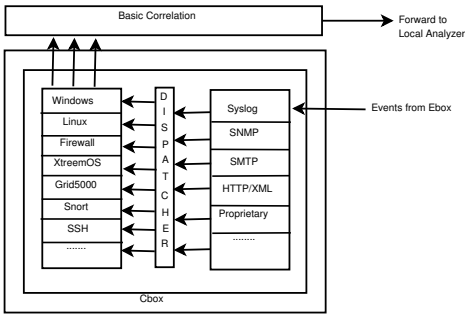

Fig. 3: CBox Design

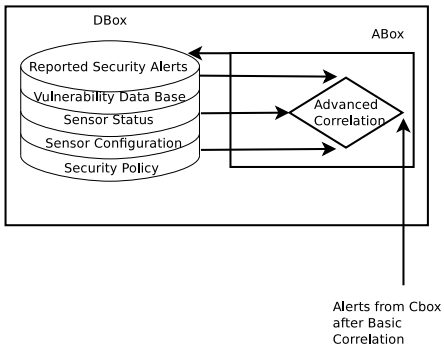

Fig. 4: DBox and ABox Design

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 strong brute force, strong ping of death and distributed denial of service attacks. The ABox warns the grid administrator with low, medium, and high-level alerts. These three types of alerts are created by the administrator using the GUI of the GSOC. These alerts are then saved in the DBox. The *DBox* holds information like *Security Policy* which contains all the rules created by an administrator for example password cracking attempts, administrative rights gaining attempts, log erasion etc. *Sensor Configuration* which holds all the information related to a node, for example what type of operating system is used on a node, its kernel version number, which services are running. *Sensor Status* shows whether the node is working or not. *Vulnerability Database* which holds vulnerability from common vulnerabilities and exposures [13]. *Reported Security Alerts* are the alerts which are identified as attacks and these alerts are saved permanently in the database.

3.4.1 Advance Correlation

The right side of figure 2 explains the advanced correlation at the LA, more specifically the ABox. When the CBox starts sending messages to the ABox, the listener module at the ABox accepts the correlated messages from the CBox. When a correlated message arrives at the ABox, a rule manager checks if the network administrator is interested in monitoring information about the sensor included in the message. If this is true, the message is stored in the lidb database and reported to the administrator. However, the administrator still does not know whether there is a strong attack on any of the sensor. For this reason, the messages that contain an alarm for an attack in itself are also stored in a aboxlocal database for a short period of time (at most one minute, just like the local database at the CBox) until the advanced correlation finishes its task. The messages without alarms are dropped, because the administrator has already been informed. 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. If there are more than one attackers that assaults the same target (sensor) in the same unit of time, a strong attack alert is stored in the lidb database. At last one alert will be displayed at the GUI of the GSOC.

3.5 Global Analyzer (GA)

The Global analyzer is the backup of the LA and lidb. It starts working if the LA and lidb are under an intensive distributed denial of service attack and if the LA stops processing security alerts from CBoxes.

4. Experiments

In this section the comparison of the GSOC with Snort and the DSOC under different network attacks has been discussed. Here Snort has been taken to measure the attack detection capability of the GSOC. In practical Snort could be used as an input to the GSOC. The DSOC has been taken to measure the performance in terms of number of alerts generated when the victim is under attack. Graphical representations have been used to show the efficiency of the GSOC in relation to others. To calculate the efficiency the number of alerts generated individually by the GSOC, DSOC and Snort in one hour has been taken as a parameter. Figure 5 is the general diagram of Grid'5000 network where the GSOC was deployed for the experiments, the details of the network are available at [14]. The deployment of GSOC for experiments was first a CBox at Orsay, a second one at Grenoble and a third CBox at the Rennes site. These three CBoxes send their logs to the LA when the two attackers simultaneously start attacking the victim machine at the Rennes site. The security alerts of the victim machine were forwarded to the LA at the Nancy site where the administrator could take action to block the attackers. In the experiments some of the log messages 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.

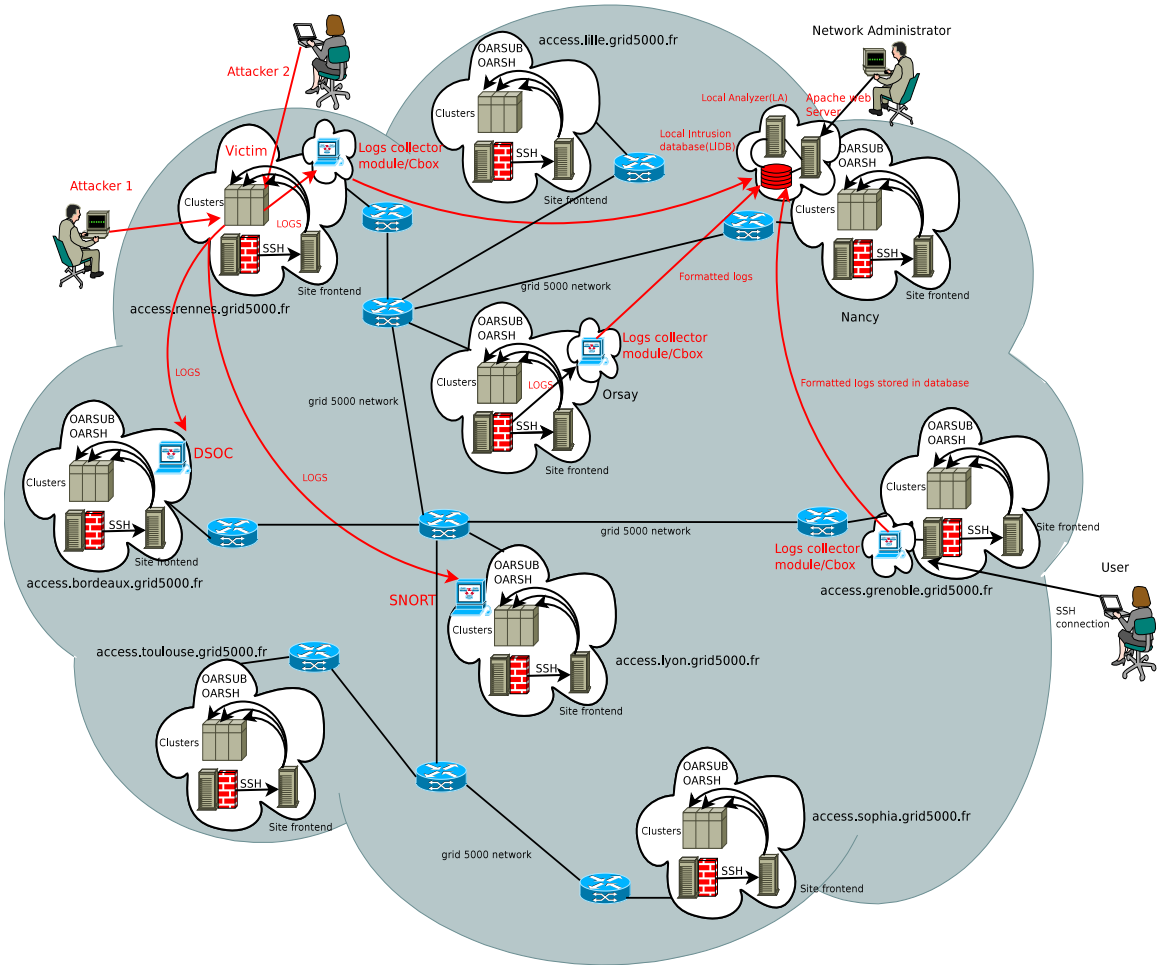

Fig. 5: GSOC in Grid'5000 Network

4.1 GSOC Behavior Under Brute Force Attack

This is a brute force attack test scenario for the GSOC. The THC Hydra [15] has been used for launching a brute force attack. It has used a password file, which is a dictionary of passwords. This dictionary contains 8048 passwords in one file. In this test, two attackers (attacker1 and attacker2) are performing the attack (see figure 5) on a target machine called victim. The GSOC detects the attack and generates one alert after one minute which has been shown on the GUI of the GSOC as a *weak attack*. This one alert contains 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 which the attack has been launched, target IP addresses and the number of attempts made by each attacker. This information is very helpful for the network administrator to stop the expansion of attacks. This is the role of basic correlation that has been added to the GSOC which minimizes the log messages to save disk space, minimizes database size, minimizes network bandwidth before sending these correlated alerts to the LA. The behavior of GSOC under brute force attack can be seen in figure 6. The LA receives the correlated messages from multiple CBoxes and further correlates them to see if the other sensors from other sites are also targeting the same sensor or a group of sensors. If this is the case then another alert is generated and displayed at the GUI of the GSOC which shows *strong attack*. This alert also contains the similar information mentioned above which helps the administrator to look further into other sites to detect the source of the attack.

4.2 Comparison of GSOC with Snort and DSOC Under Brute Force Attack

Figure 7 shows that the GSOC is much better than the DSOC and Snort. GSOC basic and advance correlation has reduced the number of generated alerts while keeping the same information intact. The GSOC has reduced the disk space, the database size and the network bandwidth; in addition, it can detect more sophisticated attacks.

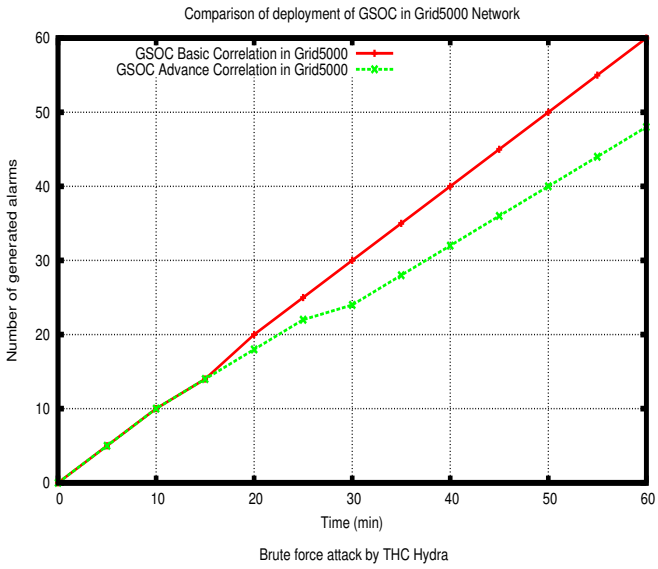

Fig. 6: GSOC behavior under brute force attack

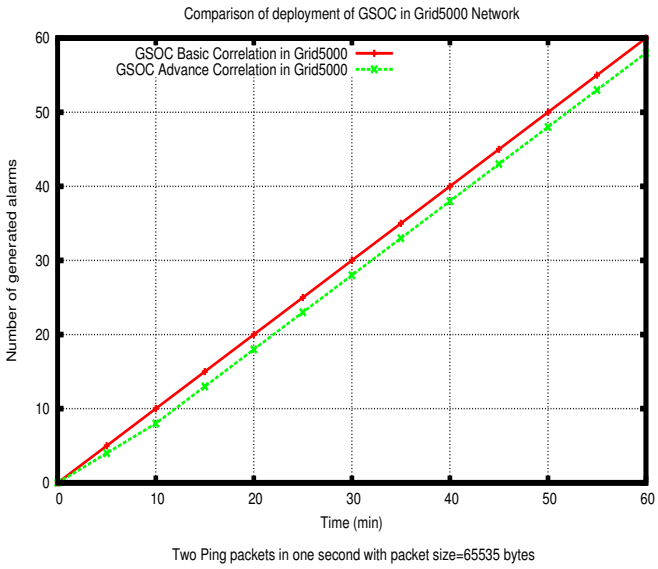

Fig. 8: GSOC behavior under Ping of Death Attack

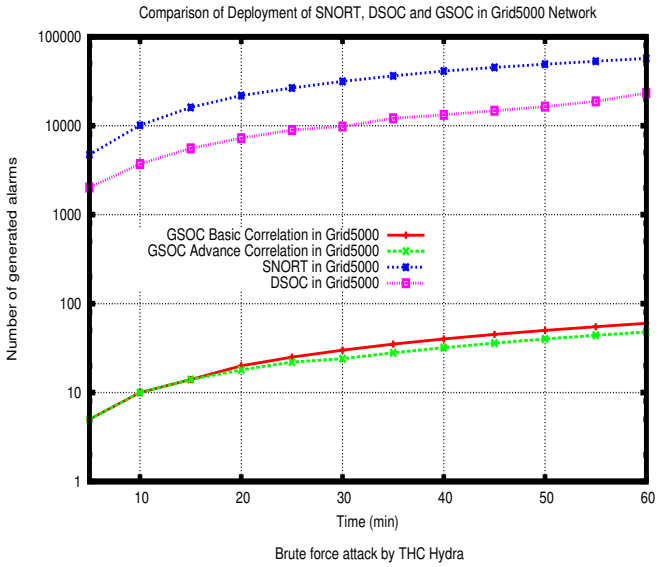

Fig. 7: GSOC v/s Snort and DSOC under brute force attack

4.3 GSOC Behavior Under Ping of Death Attack

This is a ping of death (PoD) attack test scenario for the GSOC. In order to detect a DDoS/DoS attack, ping packets bigger than (the size) of 85 bytes will be discarded. IPtable rules have been used to log an alert if any packet bigger than 85 bytes has been received by the sensor. In the code of the GSOC when the CBox script executes the IPtables, rules have been added automatically . The IPtable rules that have been 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.

This means that each ICMP packet greater than 85 bytes will be reported. After (performing) the attack using the commands mentioned below,

Attacker 1: ping -i 0.5 -s 65507 IP Address of the Victim
Attacker 2: ping -i 0.5 -s 65507 IP Address of the Victim
the results can be seen in figure 8.

4.4 Comparison of GSOC with Snort and DSOC Under Ping of Death Attack

The comparison clearly shows the performance of the Snort, DSOC and GSOC. The GSOC is much better in terms of generating lesser number of alerts. The alerts that are stored in the local database at the CBox are deleted after one minute and only the correlated messages are stored locally and transferred to the LA. This helps to control the size of the disk and database. (See figure 9).

5. Conclusion

To counter attacks like brute force, denial of service and distributed denial of service which have been discussed in this paper, the GSOC has generated few alerts compared to the DSOC. The GSOC minimizes and correlates security alerts and gives the administrator a concise and accurate security report. The results in comparison to the DSOC and the Snort are presented in section 4. The graphs in the

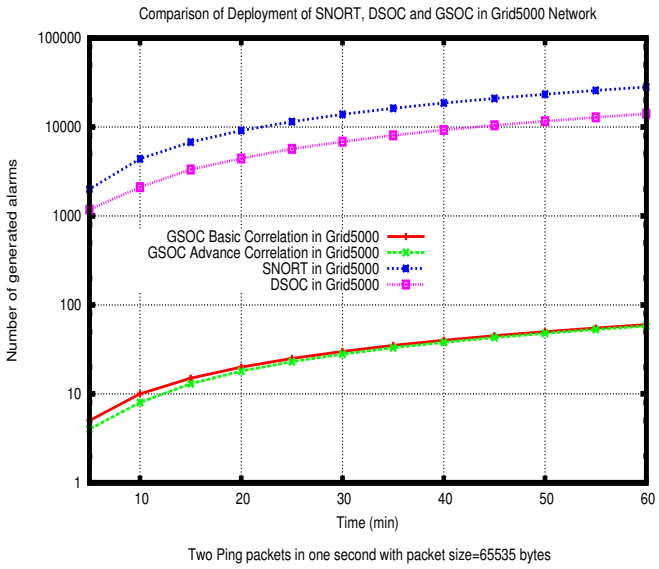

Fig. 9: GSOC v/s Snort and DSOC under Ping of Death Attack

experiments show that the GSOC generates accurate security alerts after correlating a comprehensive event record from one or multiple attackers. This correlation of security alerts makes the GSOC resistant to intensive distributed denial of service and brute force attacks.

Acknowledgments

Thanks to the Laboratory of Computer Science University of Franche-Comte France, the Higher Education Commission and Quaid-e-Awam University of Engineering, Sciences and Technology Pakistan for supporting our work financially. The Grid'5000 network for providing us with the platform to perform tests.

References

[1] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, S. Dubus, and A. 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*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 83–91. [Online]. http://portal.acm.org/citation.cfm?id=1632709.1633494

[2] Y. Xiang and W. Zhou, "Protect grids from ddos attacks," in *GCC*, 2004, pp. 309–316.

[3] "Scalable simulation framework (ssf): A public-domain standard for discrete-event simulation of large, complex systems in java and c++," 2010. [Online]. http://www.ssfnet.org/homePage.html

[4] V. Welch, J. Gawor, C. Kesselman, S. Meder, and L. Pearlman, "Security for grid services," in *In Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*. IEEE Press, 2003, pp. 48–57.

[5] I. Foster, C. Kesselman, J. M.Nick, and S. Tuecke, "The physiology of the grid: An open grid services architecture for distributed systems integration," 2002. [Online]. http://www.globus.org/alliance/publications/papers/ogsa.pdf

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

[7] R. Ford, M. Bush, and A. Bulatov, "Predation and the cost of replication: New approaches to malware prevention?" *Computers & Security*, vol. 25, no. 4, pp. 257–264, 2006.

[8] H. J. Wang, C. Guo, D. R. Simon, and A. Zugenmaier, "Shield: vulnerability-driven network filters for preventing known vulnerability exploits," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 193–204, August 2004. [Online]. http://doi.acm.org/10.1145/1030194.1015489

[9] A. K. Ganame, J. Bourgeois, R. Bidou, and F. Spies, "A global security architecture for intrusion detection on computer networks," *Computers & Security*, vol. 27, no. 1-2, pp. 30–47, 2008. [Online]. http://dx.doi.org/10.1016/j.cose.2008.03.004

[10] J. Bourgeois and S. R. Hassan, "Managing security of grid architecture with a grid security operation center." in *SECRYPT'09, Int. Conf. on Security and Cryptography, Milan, Italy*. INSTICC Press, July 2009, pp. 403–408.

[11] S. Northcutt and J. Novak, *Network Intrusion Detection*, third edition ed., ser. ISBN: 0-73571-265-4. New Riders, 2002, september.

[12] J. Bourgeois, R. Bidou, and F. Spies, "Towards a global security architecture for intrusion detection and reaction management," in *Proc. of the 4th Int. Ws. on Information Security Applications, WISA 2003*, ser. LNCS, K. Chae and M. Yung, Eds., vol. 2908, Jeju, Corea, August 2003, pp. 129–142.

[13] "Common vulnerabilities and exposures is a dictionary of publicly known information security vulnerabilities and exposures," 2010. [Online]. http://cve.mitre.org/

[14] "Grid'5000 is a scientific instrument for the study of large scale parallel and distributed systems." 2010. [Online]. https://www.grid5000.fr/mediawiki/index.php/Grid5000:Home

[15] V. Hauser, "The hacker's choice, a very fast network logon cracker which support many different services," 2010. [Online]. http://freeworld.thc.org/