***I.J. Information Technology and Computer Science*, 2016, 3, 75-84**
Published Online March 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.03.09

Immune-Inspired Self-Protection Model for Securing Grid

Inderpreet Chopra
ADM, Expicient Software Pvt. Ltd., India
E-mail: inderpreet20@gmail.com

Ratinder Kaur
Research Scholar, Thapar University, Patiala-147004, India
E-mail: ratinder@thapar.edu

***Abstract*—**The application of human immunology in solving security problems in Grid Computing seems to be a thought-provoking research area. Grid involves large number of dynamic heterogeneous resources. Manually managing the security for such dynamic system is always fault prone. This paper presents the simple immune based model for self-protection (SIMS) of grid environment from various attacks like DoS, DDoS, Probing, etc. Like human body helps to identify and respond to harmful pathogens that it doesn't recognize as “self”, in the same manner SIMS incorporates the immunological concepts and principles for safeguarding the grid from various security breaches.

***Index Terms*—**DDoS, Immune System, Security, Snort, Globus.

I. Introduction

Grid infrastructure provides us with the ability to dynamically link together resources as an ensemble to support the execution of large-scale, resource-intensive, and distributed applications [1]. With the evolution of Grid, the complexity of the distributed systems has increased and therefore the implementation of a secure environment has become difficult. The real power of grid can be harnessed only if it provides secure access to resources/services. The grid security is a multidimensional problem [2]. There are several factors that make security hard e.g. user population and resource pool is large and dynamic, resources have different authentication and authorization requirements, computations span over multiple domains, users have different roles/privileges in different domains etc. [11]. All these factors make security a big and challenging issue. According to [27], grid security solutions are categorized as, System Solutions, Behavioral Solutions and Hybrid Solutions. System solutions manipulates hardware and software of grid directly to achieve security. These solutions are centered upon system based security for grid resources and IDS. Behavior solutions emphasize policy and management controls. It addresses all accountability, group management and trust related issues. The hybrid solutions handles Authentication and Authorization concepts for grid users. This paper mainly deals with System level solutions. It is found that, most problems stem from the fact that human administrators are unable to cope up with the amount of work required to properly secure the computing infrastructure at the age of Internet [3]. Autonomic systems and self-managed environments provide security policies establishing promising trust between the server and the client. Hence, autonomic computing [4, 5] presented and advocated by IBM, suggests a desirable solution to this problem [6]. Self-Protection enables the system with an ability to secure itself against attacks i.e. detect illegal activities and trigger counter measures in order to stop them. It also helps to overpower the limitations of manual management of the system i.e. reduced speed, increased chances of errors and un-manageability by human administrators. This paper discusses a human immune system based self-protection model for grid. Like the human immune system protects the cells from the invasion of pathogens and viruses, our proposed model uses the same concept to safeguard the grid from various network attacks. Our system has the strong abilities of learning, recognition and characteristics extraction.

1.1 Self-Protection

A self-protecting system helps to detect and identify hostile behavior and take autonomous actions to protect itself against intrusive behavior. The main goal of self-protection system is to defend grid environment against malicious intentional actions by scanning the suspicious activities and react accordingly without the user's awareness that such protection is in process [7].

The main design principles required to build a self-protected [7] system are summarized below:

- A self-protected system must be able to detect intrusions. It requires a definition of its own operations: this is the sense of self capacity or the self-knowledge aspect. In other words, it must be able to distinguish legal behaviors from illegal behaviors. As the countermeasures are triggered autonomously, this distinction must be done while avoiding false positives. Moreover, the legal operations or the system structure could evolve over time: as a consequence, self-knowledge requires dynamic introspection capabilities.

- The system must have the ability to respond to attacks. This capability relies on the capacity for the system to reconfigure all its individual components.
- A wide variety of systems must be protected, including legacy software not designed to be autonomic.

The components involved in the self-protection of the system can also become a target of attack. If somehow, the components got compromised, the attackers can use them in an unintended way leading to negative consequences. Hence, the system must prevent the self-protection components from being compromised.

II. Related Work

This section deals with research focused on system based solutions toward grid security. Proposed solutions falling into this category seek to protect resources on the grid. Thus focuses on protecting the grid resources, which include grid nodes (Host) and communication network. Several isolation techniques such as Sandboxing and Virtualization comes under this category to protect grid nodes [28]. The Entropia (called the Entropia Virtual Machine) system uses a technique known as sandboxing to protect applications, clients, processes and resources on the grid [29]. EVM has been specifically designed to cater to the desktop grid environment, where there are a large number of desktop clients on which the grid jobs run, in addition to the Entropia server. Another way to provide isolation is through Virtualization where an illusion of a single machine is provided through the creation of Virtual Machines. In [30], the Virtual Private Grid infrastructure (VPG) is proposed. This involves harnessing virtual private network technology and applying it to grid computing. This infrastructure works around heterogeneous, locally-specific security. Most of these attacks are kind of distributed denial of service attacks (DDoS) [39]. Attempt to provide security against DDoS on Grid is made by Varalakshmi et.al.[37]. They have proposed a five-fold DDoS defense mechanism using an information divergence scheme. This detects the attacker and discards the adversary’s packets for fixed amount of time in an organized manner.

Intrusion detection Systems (IDSs) [38] is another way to provide system level security. IDSs add an early warning capability for the suspicious activity that mostly occurs before and during an attack. Intrusion detection systems (IDSs) often work as misuse detectors, where the packets in the monitored network are compared against a repository of signatures that define characteristics of an intrusion. In another approach, the anomaly based intrusion detection is the method for detecting intrusions by monitoring the packets for any anomalous behavior. This classification is based on heuristics or rules rather than patterns or signatures [34]. Forrest, et. al. is one of the pioneering researchers in using the evolving models of immune system in solving the optimization problems [35]. Recently, an agent-based IDS (ABIDS) inspired by the danger theory of human immune system is proposed. Multiple agents are embedded to ABIDS, where agents coordinate one another to calculate mature context antigen value (MCAV) and update activation threshold for security responses [36].

Many grid based IDS systems have also been conceived, designed and implemented. [9] The basic components for grid based IDS systems include: Sensors, which are able to monitor the state of grid systems. The information collected by sensors are then collected and analyzed by IDS system like SNORT [10]. Based upon the analysis, alarm is raised. Many approaches for Grid based IDS prevailed in the market. Kenny and Coghlan [12] describe a system that allows the querying of logfiles through the Relational Grid Monitoring Architecture (R-GMA), which can be used to build a Grid-wide intrusion detection system. Their implementation of this system, called SANTA-G (Grid enabled System Area Networks Trace Analysis), queries Snort log files by using SQL. SANTA-G is composed of three elements: A Sensor, a QueryEngine and a Viewer GUI. Schulter et al. [13] describes different types of IDS systems. The authors point out that present Grid-IDS approaches do not fulfill the important qualities (completeness - recognition of all attack-types, scalability and Grid-compatibility) for protection of Grid systems. They propose a high-level GIDS that utilizes functionality of lower-level HIDS and NIDS provided through standard inter-IDS communication. Sarkar S. and Brindha. M [40] presents the detail on other NIDS systems available.

Another example of grid based IDS is GIDA [31] which also uses a similar structure. IDS on Oracle lOG database is provided in [32]. IACID [33] from USC, provides a Grid based IDS system having separate network and host IDS systems. Michal Witold [14] in his thesis investigated the possibility of Grid-focused IDS. The main stress has put on feature selection and performance of the system. Leu and Li [15] proposed Fault-tolerant Grid Intrusion Detection System (FGIDS) which exploits grids dynamic and abundant computing resources to detect malicious behaviors from a massive amount of network packets. In FGIDS, a detector can dynamically leave or join FGIDS anytime. Intrusion detection system must analyze a large volume of data while not placing a significant added load on the monitoring systems and networks. There are several limitations of existing IDS discussed above: current systems report alerts to a centralized database and hope that a network administrator is immediately on-hand to take an action. The administrator analyzes alert reports and takes some actions. Other proposed system analyze database using knowledge discovery techniques for deciding which actions must be taken. It’s often too late in critical systems to decide where actions must be taken by one centralized device or user when the given system may have been compromised and vital information may have been stolen.

III. Artificial Immune System

Medical field refers "immunity" to the resistance exhibited by the hosts towards injury caused by microorganisms and their products. Immunity in human beings is provided by a complex network of organs and cells responsible for defense against alien particles called the "human immune system". This system comprises of several layers of defense: The first layer comprises of body's external defenses that include the tissues which cover and line the body. For example unbroken skin providing barrier from invading pathogens and mucous membranes lining the respiratory track preventing entry of pathogens [21]. Second and Third layers of defense comprise the innate immune system and adaptive immune system.

3.1 Innate Immune System

It comprises specialized cells and molecules which initiate response to any pathogens that enter the body. This is achieved through special receptors on the surface of innate immune system cells. These receptors are germ line encoded, consequently the innate immunity is "built-in" from birth and doesn't change with growth or experience.

3.2 Adaptive immune system

This system initiates a response specific to pathogens that has entered the body. It also provides memory capabilities to immune system. Lymphocytes play a central role (Figure 1). They circulate through the blood and lymphatic system waiting to encounter antigens (The foreign molecules belonging to pathogens that invade the body) in adaptive immune system. These are classified into two main types: B-Cells and T-Cells. These lymphocytes begin as the stem cell found in the bone marrow and go through a maturation process. B-Cells complete their maturation by entering into the bursa of fabricus forming immune competent "bursal lymphocytes" or B-Cells (B for bursa or bone marrow), while T-Cells move to the thymus to complete maturation forming T-lymphocytes.

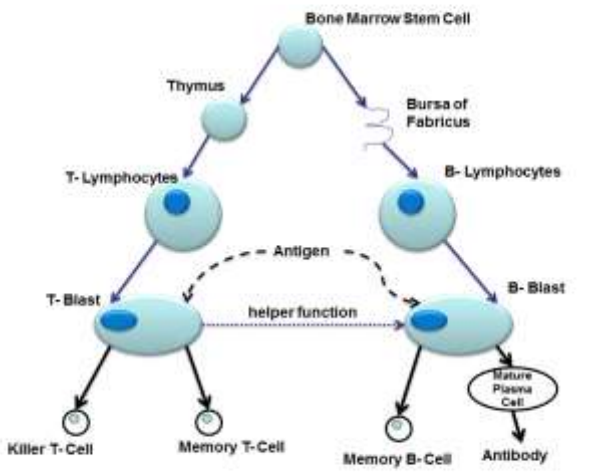

Fig.1. Origin of T and B Cells

One of the important processes that T-lymphocytes undergo is that of central tolerance involving the elimination of lymphocytes that recognize self-antigens and would otherwise initiate an immune response to self (autoimmunity). This provides immunological tolerance towards self, implemented as "Negative selection". T-Cells may be broadly classified into regulatory and effector cells. Regulatory T-Cells are the helper T-Cells or suppressor T-Cells. Effector cells are the killer T-Cells.

B-Cells on the other hand undergo clonal selection or energy. It is the process in which B-Cells binding to the more specific antigens have more chance of being selected for cloning. Whenever an antigen attacks, majority of the activated B-Cells are transformed into plasma cells which are the antibody secreting cells. While few of the activated B-Cells, they produce memory cells for the reoccurrence of same antigens in future. Therefore the secondary response is quicker.

The crossover between biology and computer science can be fruitful. As the Human Immune System protects the cells from the invasion of pathogens and viruses, artificial immune systems defends against the intrusion outside and inside: both of the system make themselves stable in constant changing environment [20].

IV. Simple Immune based Model for Self-Protection (SIMS)

We have discussed the various problems and the related work that has already been done to avoid the grid system services to be compromised. All of these areas are under constant investigation by researchers and have been so for a long time. There is still no grid middleware that is intelligent enough to handle the attacks and faults automatically. Simple Immune based model for Self-protection (SIMS) approach is inspired from immune system and provides the agent based self-management environment to manage the security in grid. Without external maintenance or management, SIMS automatically detects known intrusions and auto generates the signatures to handle the new attacks.

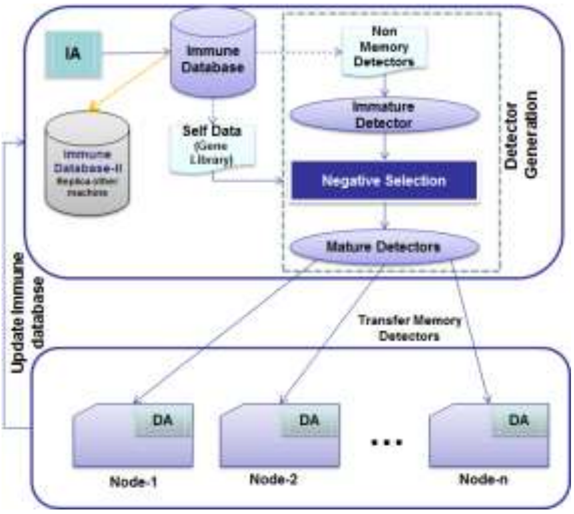

Fig.2. SIMS Architecture

The overall architecture of SIMS is presented in Figure 2. SIMS consists of an Immune Agent (IA) running on root node and the various desktop agents (DA's) running on child nodes. IA is similar to Bone Marrow/Thymus that generates lymphocytes and IA generates numerous detector sets. Detector sets are based upon the negative selection algorithm. Negative selection helps to generate the diverse detectors, which do not match self-data. Lymphocytes keep on monitoring the body for pathogens (antigens) and in the same manner the detector sets generated by IA's are used to monitor the whole network for any kind of network attacks. Each detector set describes abnormal patterns of network packets and is transferred to all DA's. In DA, detectors keep on running as the background process which monitors self from non-self-traffic patterns as observed from the network traffic patterns.

Bone Marrow →Lymphocytes
SIMS **Immune Agent** (IA) → SIMS **Desktop Agents** (DA) {detector sets}

As discussed in above section, Lymphocytes comprises of two kinds of cells: B-Cells and T-Cells. B-cells are usually present from the birth whereas T-Cells keep on adding into the system as new antigens are detected. T-cells are further subdivided into two classes, T-helper cells and T-killer cells.

Role of helper cells is to invoke the existing processes (usually B-cells) to handle the antigen by producing the antibodies for the same. On other hand T-killer cells try to generate the new set of antibodies to handle antigens if encountered in future. SIMS has used the same concept of B-Cells and T-Cells. DA consists of existing detectors (equivalent to B-Cells) and new detectors (equivalent to T-Cells). Existing detectors have the initial rule set that handles the already known attacks. New detectors perform two functions- protect the environment from existing known attacks and raise alerts whenever any intrusion is detected. Table 1 summarizes the SIMS components and there brief description with reference to immune property in immune system.

Lymphocytes → {**B-Cells**, **T-Cells** [helper, killer]} present in body with blood
SIMS **DA** → {**existing Detectors**, **new Detectors** [protect, alert]} present as independent agents running on every node

Table 1. Immune Terms and their Meaning in SIMS

Immune Property	Description	SIMS	Description
Bone Marrow	Produce white blood cells that contains lymphocytes in high quantity	Immune Agent(IA)	Main agent that generates different detector set definitions and manages DA's.
Lymphocytes	Lymphocytes function as small independent detectors that circulate through the body in the blood and lymph systems	Desktop Agents(DA)	DA's are running on each node participating in grid environment
Antigens	An antigen is a foreign substance that triggers a reaction from the immune system	Network Attacks	Attacks like DoS, DDoS, Probing that can harm the system
Antibody	A kind of special proteins that mark intruders for destruction by other cells.	Detector Set	Signatures/Rules for snort to handle the attacks if the same attack occurs again
Self/non-Self	molecules and cells of the body (called "self") and foreign ones (called "nonself")	Known frequently occurred /unknown patterns	Self is considered as the data packets those are known to the system based upon the authentication of user. Non-Self are the unknown data packets that can cause trouble to the grid environment.
Negative Selection	negative detectors, because they detect non-self-patterns, and ignore self-patterns	Detect non-Self from self-data	Done at IA to generate new detector set.

The key differentiators of SIMS are:

- **Distributability**: Lymphocytes in the immune system are able to determine locally the presence of an infection. DA's in SIMS can work as an independent entity and are able to detect attacks based upon their local detector set definitions. No central coordination takes place, which means there is no single point of failure. This distributed architecture even makes the system more robust and useful for grids. Most of the traditional systems we have found are usually centralized and thus there is always a single point failure risk.
- **Automated Protection**: The immune system is autonomous, regenerating damaged cells and classifying and eliminating pathogens, all without outside intervention. Similarly, SIMS try to automate the signature generation process for IDS based upon different alerts raised by different DA's. This is our novel approach to self-protect

the system from attacks. The detail on this is mentioned in section 5.1.2.

- **Automatic Push Updates**: Like lymphocytes are always getting updates from bone-marrow based upon the new antibodies detected, similarly the set of detectors present inside the DA's get detector updates for the new attack for which alert has been raised.

SIMS working is described in three steps:

4.1 Gather the Self-Data

The method of gathering the self-data was implemented in two phases: an initial phase, and an actual phase. During the initial phase, a database of normal behavior was collected for each grid service of interest. The database was collected by adding the default grid services information and the authentic user details. Self-database was specific to a particular system (defined by its configuration, architecture, software version, etc.). In the actual phase, the running grid was monitored for deviations from the recorded normal behavior, that is, the system scanned for occurrences of sequences not in the normal database for that program.

Self_Data

sid: NSID
content: NCONTENT

CONTENT: {IP, PORt, USER_DETAILS}

Self-acts as the antibody and is the root class for our model. It is determined by an identifier sid which belongs to a set of identifier SID. This set is a subset of N. Each antibody has various distinct properties that are represented as content, it contains properties like IP, Port various user details like its mac information, secret pass key, depth etc.

Database Design for our model is divided into three categories: Snort Database, Self-Database and Acid Database. Snort database store the data related to snort. It logs the snort detected intrusions information. Self-Database contains the information related to the self and non-self-information. Based upon the information in the self-database, the rules are auto generated which are later transferred to desktop agents. Acid database is just for the monitoring purpose.

Immune_DB

snortDB
selfDB
gridDB

snortDB: {Logs, Nodes_Config}
selfDB: {Self, Immature, Mature, Memory}
gridDB: {Monitoring}

4.2 Generate Random Detectors

IA reads the alert table data and after some processing, generates the rules in rules table. SIMS first reduces the redundancy in data by merging the alerts and removing the duplicate alerts. The rule for merging is: if any of the two alerts that source IP, destination IP and IP version are same, then the two would integrate as one and other attributes like ip_len, ip_tos, etc. are merged and named as immature detectors. After merging of redundant data is done, IA first compares the immature detectors with the self-string from self-DB. If the entry is not there, immature detector is converted to mature detector. This rule is added to the rules database from where the new detector set is generated. This is approach is similar to the Negative selection approach of the Immune system. Negative selection is a mechanism employed to help to protect the body against self-reactive lymphocytes to avoid autoimmunity.

Detector

Rule_Header: P{IP, Port}
Rule_Option: P{Content, Depth}
Rule_SID: NSID

Rule_SID C SID
Rule: {Rule_Header, Rule_Option, Rule_SID |
Rule → Rule_Sid= φ}

The algorithm we use in this is:

- Define self as a collection of strings S of length l over a finite alphabet.
- Generate a set of R detectors, each of which fails to match any string in S.
- Monitors for changes by continually matching the detectors in R against S. If any detector ever matches, then the change is known to have encountered as detectors are designed not to match any original string.

4.3 Formation of Memory Detectors

As the DA's detects any intrusion, they logged the information in the centralized database monitored by the IA. As more and more intrusions are reported by the different DA for the particular intrusion, the affinity factor for that DA increases. When this affinity factor reached the threshold value, the details of the intrusions are moved from the non-self-table to self-table and the rules are generated for that particular entry.

The detail algorithm for SIMS is presented in Appendix.

V. Implementation and Results

Figure 3 shows the complete view of SIMS implementation setup. Following sections explains the implementation process of SIMS and its results.

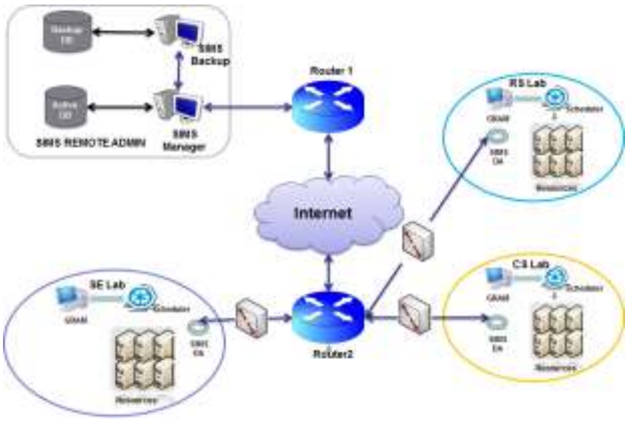

Fig.3. Implementation Setup

5.1 *Software's Used*

The implementation uses: standard IDS called Snort, ACID monitoring tool, apache web server to host ACID, MySQL database and Java SDK6 for programming the tool.

5.2 *Grid Environment*

Grid Environment setup using Globus Toolkit 4 [23]. The major components involved are:

- GRAM: Enables resource allocation through job submission, staging of executable files, job monitoring and result gathering [24].
- Scheduler: Helps to schedule the jobs to resources available. Here we use the GridWay Meta Scheduler [25] to schedule the jobs.
- Grid FTP: Extension of standard FT protocol that provides the secure, efficient and reliable data movements in grid environment [26]. In addition to standard FTP features, it provides GSI support for authenticated data transfer.

5.3 *Experimental Setup*

Snort is used as the base tool for implementing the immune based security model. Snort is used to leverage existing distributed IDS capability to one step ahead by automating its rules generation process by the use of Immune system. For testing the immune model, we installed snort sensors on different resource nodes participating in the Grid environment, then installed the database (MySQL) on one machine, and deployed a Web server and ACID onto another machine. The grid environment is setup inside the Thapar University (TU) campus using GT4 middleware. Grid environment consists of 44 Intel Dual Core 2.2GHz processor Windows XP nodes, 20 dual 2.4GHz Xeon Linux nodes and 5 nodes dual 450MHz PII Linux clusters. Each node has 1GB RAM and 80GB HDD (Table 2). TU Grid is exposed to the outer world with limited access.

Table 2. SIMS Environment Details

Configuration	OS	Number
Intel Dual Core 2.2GHz	Windows	44
2.4GHz Xeon	Linux	20
450MHz PII	Linux	5

5.4 *Results*

We have implemented the model in two phases- Initial phase and the Actual phase. The security system was tested, using data gathered at the TU. The data is gathered from 69 odd computers in grid environment from different labs using tcpdump utility by monitoring the network for 1month, to yield a total of nearly 1 million data chunks. Attacks were launched under controlled experimentation over the WAN link. Metasploit framework was used to launch various attacks. The Metasploit framework is open source software for use in performing penetration testing, intrusion detection prevention system signature development, and exploit research.

Unwanted data (data that is not grid specific) is filtered out from 1 million data chunks and we are left with 0.21 million strings. These data were used for both- initial and actual self-data sets, and non-self-actual sets. All non-self-traces were manually extracted from the data, thus separating the self and non-self-patterns. From the above gathered data, 0.18 million strings are used to form the self-data. Out of 0.18 million, 2900 unique strings are chosen and rules for these are generated. The rules impose conditions on various GT4 components, these includes port check on GRAM (2119), MDS (2135), Grid FTP(2811), GSI SSH(22) over the TCP protocol for the default users. Rest of the 0.03 million strings are used as the non-self-data strings. These acts as the data for detectors generated from the self-data.

Table 3. Example Alerts Raised

sid	Ip_src	Ip_dst	Ip_ver	other
1	3232235777	3232235778	4	{192},{576},{0}
2	3232235779	3232235778	4	{32},{576},{0}
3	3232235777	3232235778	4	{32},{576},{0}
4	3232235777	3232235778	4	{192},{60},{0}
5	3232235779	3232235778	4	{0},{576},{0}
6	3232235777	3232235778	4	{0},{576},{0}
7	3232235775	3232235778	4	{32},{576},{0}

Table 4. Example Compressed Alerts database to be used in Rule formation

rid	Ip_src	Ip_dst	Ip_ver	other
1	3232235777	3232235778	4	{192,32,0},{576,60},{0}
2	3232235779	3232235778	4	{32,0},{576},{0}
3	3232235775	3232235778	4	{32},{576},{0}

To understand this process, let's take a small chunk from gathered data. Table 3 shows alert database formed from the information obtained by the alerts raised by snort based desktop agents. From this database, the unique strings are chosen to form the rules database Table 4. The algorithm for the same process is discussed in Appendix.

The experiments were conducted by considering two categories- Traditional and SIMS. The difference between the two is that the traditional system lets the snort itself to generate new rules definition (managed manually by some person) while SIMS automates the rules generation process.

The results in the Figure 4 points out that the traditional method complete its work rapidly while alert number is fewer. But as the alerts count increase, the time to analyze the rules increases for traditional system. This is because of the large data in the rules database. But in case of SIMS, we are merging and discarding the unused rules, thus reducing the number of rules in the database. Figure 5 plots the current count of rules produce by both the traditional and SIMS system. SIMS generated rule count is very less as compared to Traditional method. Still in Figure 6, we can see that the success rate for the attack handling is better in SIMS.

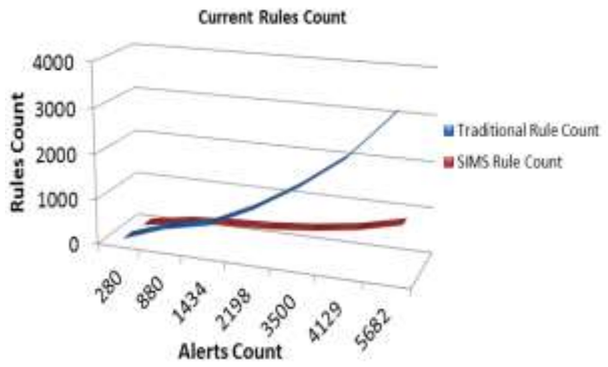

Fig.5. current rules count

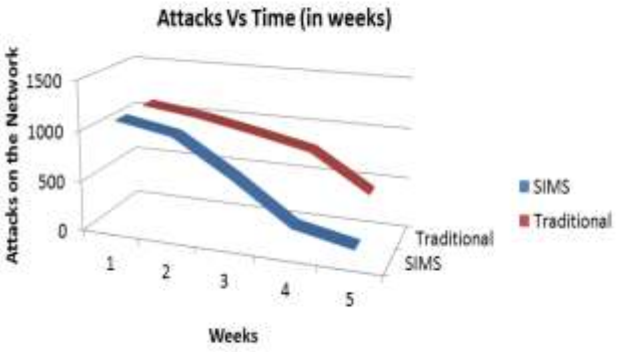

Fig.6. Attacks Vs. Time (in weeks)

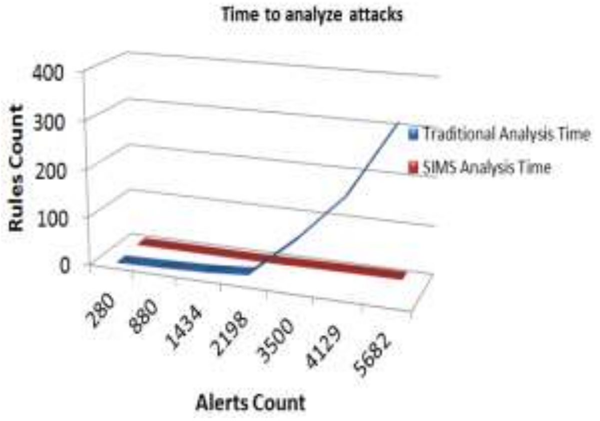

Fig.4. Time to analyze attacks

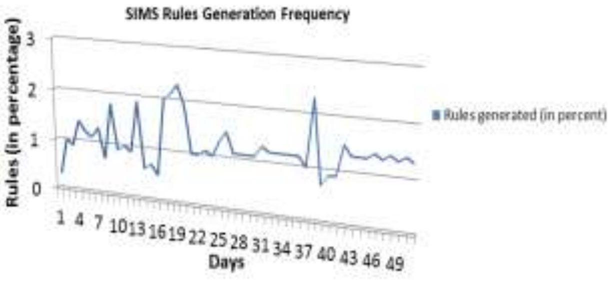

Fig.7. SIMS Rules generation frequency

Table 5. Detection Rate and False Positive Rate

(SIMS/ Traditional)	Week 1		Week 2		Week 3		Week 4		Week 5	
Detection rate (%)	51	79	59	81	75	83	87	88	97	91
FP Rate (%)	90	50	66.3	57	45.7	59	32.1	55	28	61

As the time passed, the effectiveness of rules is decided on the basis of the alerts raised by it. If the rule raises the correct alert and is in use, its impact value is increased and vice-versa. The rules are represented by SID and their impact value is called the affinity. More is the affinity, higher is the priority. Figure 7 shows the rules generation frequency of SIMS.

To further verify our system, we have used two standard metrics- Detection Rate and False Positives rate to support our system. Detection rate is the number of attacks detected and blocked whereas false positive rate is the ratio of invalid alerts to the true alerts for the true traffic. We can see in the Table 5 that with the increase in time, the percentage of false positive reduces and the detection rate increases for SIMS. Here we have shown only data for 5 weeks as this period we consider as the training period for SIMS.

It was observed that with the passage of time, security of the grid environment with SIMS keeps on increasing. And as the time passed, the number of rules auto generated with the help of genetic features, crossing over and mutation also increases. It was also observed that the

SIMS kept on adding the rules based upon the alerts raised into the database for the next few days.

VI. Conclusions

In Grid Computing, the resources belong to different administrative domains and are geographically distributed, their availabilities may be very dynamic. This make grid environment increasingly complex and difficult to administrate. This complexity is such that the presence of bugs and security holes is unavoidable. In this paper, SIMS model based upon human immune system is proposed and implemented. The proposed model helps to distinguish self from non-self-operations. The central node in SIMS automatically transfers the generated rules to all the child nodes and this feature helps in case of failure of central node by still keeping the local system protected. Another advantage is that this system is easily scalable. As the central node carries all the rules, once the new node is registered with the grid system, it will automatically receive the entire existing rule set from the central node. The detailed working of SIMS has been described. The goals achieved by the proposed system are summarized as:

- Reduction of administrative complexities: SIMS reduces the manual intervention of administrators to generate and manage new rules. Administrators are mainly needed at initial level to form the self-database. Later with time this self-database keep on updating itself wrt alerts raised.
- Automatic protection of the system from malicious activities like DoS, DDoS, Probing, etc by SIMS Self-Protection mechanism. New rules are added, old unused rules are deleted and the existing rules are monitored by SIMS.
- Scalability of the system to add new nodes easily into the grid.

Appendix A: Algorithm for SIMS

```
IMMATURE DETECTOR COUNT=0;
MATURE DETECTOR COUNT=0;
MEMORY DETECTOR COUNT=0;
Do {
#Gather Self Data {
Data is collected            {
    When user is authenticated to use Grid, that data is self-data
    Existing grid services ports and running information related configurations
    }
    IF data already not in Self DB THEN
                Add data and create new SID
    ENDIF
}
```

```
# Merge Redundant Alert Data {
 Process alert DB, A={$A_1$, $A_2$,…,$A_n$}
    IS_PROCESSED is flag in each alert row to know its status
    FOR EACH A where IS_PROCESSED='FALSE'
        Select all alerts where A`={src_ip, dst_ip, ip_ver} ≈ $A_n${ src_ip, dst_ip, ip_ver}
         FOR EACH A'
            IF A'1 is equal to $A_n$ THEN:
                DELETE A'1
            ELSE
                MERGE all A'1 fields to $A_n$
                Mark A'1 as processed in Alert DB. IS_PROCESSED='TRUE'
         END FOR
        Add An→S, where S is set of unique strings S={$S_1$, $S_2$,..,$S_n$}
    END FOR
}
```

```
#Generate Immature Detector {
    FOR EACH S
        Generate the Immature detector, string of bits I=11000010000….00010100011
        IMMATURE DETECTOR COUNT++
    END FOR
}
```

```
#Form Mature Detector {
IF any I.equals (self_DB string) THEN
    Delete immature detector
    IMMATURE DETECTOR COUNT - -
ELSE
    Change immature detector as new mature detector
    MATURE DETECTOR COUNT++
ENDIF
}
```

```
#Form Memory Detector {
Check whether any mature detector detects any non-self-
  antigen
IF true THEN
  increase the affinity value
ENDIF
Check whether any mature detector detects any self antigen
IF true THEN
  decrease the affinity value
ENDIF
Convert those mature detectors whose affinity level
reaches threshold level to new memory detector
MEMORY DETECTOR COUNT++
Delete the Old mature detectors
MATURE DETECTOR COUNT--
Update the self-DB
}
```

```
#Transfer Detectors {
    FTP the new memory detectors to participating Nodes
}
```

```
#Replicate Database {
    Keep the Immune database backup after each update
}
```

```
} While (SYSTEM==RUNNING)
```

References

[1] Fran Berman, Geoffrey Fox and Tony Hey, "The Grid-past, present, future, Grid Computing Making the Global Infrastructure a Reality," *John Wiley & Son*, 2003.

[2] L. Ramakrishnan, "Securing Next Generation Grids," IT

Pro, IEEE Computer Society, 2004, pp. 34-39.

[3] I. Chopra and M.Singh, “Agent based Self-Healing System for Grid Computing,” ICWET, ACM, Feb 2010, pp. 31-35.

[4] Alan Ganek, “Overview of Autonomic Computing: Origins, Evolution, Direction,” CRC Press, 2004, pp. 3-18.

[5] Jeffrey O. Kephart and David M. Chess, “The Vision of Autonomic Computing,” IEEE Computer, 2003. Pp.41-50.

[6] Hang Guo, Ji Gao, Periyou Zhu and Fan Zhang, “A Self-Organized Model of Agent Enabling Autonomic Computing for Grid Environment,” 6th Word Congress on Inteligent Control, 2006, pp.2623-2627.

[7] Benoit Claudel, Noel De Palma, Renaud Lachaize and Daniel Hagimont, “Self-protection for Distributed Component-Based Applications,” Springer-Verlag Berlin Heidelberg, 2006, pp.184-198.

[8] Christian Kreibich and Jon Crowcroft, “Honeycomb - Creating Intrusion Detection Signatures Using Honeypots”, ACM SIGCOMM, January 2004, pp.51-56.

[9] Anirban Chakrabarti, “Grid Computing Security”, Springer, Ch-6, 2007 pp105.

[10] Snort, https://edge.arubanetworks.com/article/leveraging-centralized-encryption-snort-part-1

[11] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, “A Security Architecture for Computational Grids”, 5th ACM Conference on Computer and Communications Security, 1998, pp. 83-92.

[12] S. Kenny and B. Coghlan, “Towards a Grid-Wide Intrusion Detection System”, Advances in Grid Computing, Springer, 2005, pp. 275-284.

[13] A. Schulter, F. Navarro, F. Koch, and C.Westphall, “Towards Grid based Intrusion Detection”, Network Operations and Management Symposium, NOMS, 10th IEEE/IFIP, 2006, pp. 1-4.

[14] Michal Witold Jarmo lkowicz, “A Grid-aware Intrusion Detection System”, Technical University of Denmark, IMM-THESIS, 2007.

[15] Fang-Yie Leu, Ming-Chang Li and Jia-Chun Lin, “Intrusion Detection based on Grid”, ICCGI, 2006, pp. 62-68.

[16] Michael P. Brennan,”Using Snort For a Distributed Intrusion Detection System”, SANS Institute 2002, pp. 1-12.

[17] Hassen Sallay, Khalid A. AlShalfan and Ouissem Ben Fred j, “A scalable distributed IDS Architecture for High speed Networks”, IJCSNS, VOL.9 No.8, August 2009, pp. 9-16.

[18] Yasir Abdelgadir Mohamed and Azween B. Abdullah, “Immune Inspired Framework for securing Hybrid MANET”, IEEE Symposium on Industrial Electronics and Applications (ISIEA), October 2009, pp301-306.

[19] Youngzhong Li, Rushan Wang and Jing Xu, “A Novel Distributed Intrusion Detection Model Based on Immune Mobile Agent”, International Symposium on Web Information Systems and Applications (WISA), May 22-24, 2009, pp.72-75.

[20] Zhang Quing-hua, Zhang Ya-she,Shao Long-qui and et.al., “An Immunity Based Technical Research Into Network Intrusion Detection”, International Conference on Computer Science and Software Engineering, 2008, pp. 955-958.

[21] R Ananthanarayan and C.K.J. Paniker, “Textbook of Microbiology”, 6th ed, Orient Longman, 2000.

[22] Carranza and Newman, “Clinical Periodontology”, 8th ed, Harcourt India Pvt Ltd, pp118, 2001.

[23] Globus Alliance, “A Globus Primer: Describing Globus Toolkit Version 4”, http:/www.globus.org/toolkit/docs/4.0/, 2008.

[24] M. Feller and I. Foster and S. Martin, “GT4 GRAM: A Functionality and Performance Study”, 2008.

[25] Ruben S. Montero, “The GridWay Meta-Scheduler”, Open Source Grid and Cluster, Oakland, CA, May 2008.

[26] Data Management: Key Concepts, Grid FTP, http:/www.globus.org/toolkit/docs/4.0/data/gridftp, 2008.

[27] Erin Cody, Raj Sharman, Raghav H. Rao and Shambhu Upadhyaya, “Security in grid computing: A review and synthesis”, Decision Support Systems, 2008, pp749-764 .

[28] A.Bendahmane, M.Essaaid i, A.El Moussaoui and A.Younes, “Grid Computing Security Mechanisms: State-of-The-Art”, Support Systems, Multimedia Computing and Systems, ICMCS '09, 2009, 535 - 540.

[29] A. Chien, B. Calder and S. Elder, “ENTROPIA: architecture and performance for an Enterprise desktop grid system”, Journal of Parallel and Distributed Computing 2003, pp597-610.

[30] F. Kon, M. Roman and P. Liu, “Monitoring, security and dynamic configuration with the dynamic TAO reactive ORB”, IFIP/ACM International Conference on Distributed Systems Platforms, New York, United States 2000, pp. 121-143.

[31] M.F. Tolba, M.S. Abdel-Wahab, LA. Taha and A.M. Al-Shishtawy, “GIDA: Toward Enabling Grid Intrusion Detection System”, Proc. Conference on Cluster Computing and Grid (CCGrid), Cardiff (Wales), 2005, pp.1-3.

[32] Oracle 10g database, www.oracle.com/technology/products/bi/odm/pdf/odm based intrusion detection-paper 1205.pdf, 2006.

[33] T. Ryutov, C. Neumann and L. Zhou, “Integrated Access Control and Intrusion Detection (IACID) Framework for Secure Grid Computing”, Tech. Report., University of Southern California, 2005.

[34] K. Hwang, M. Cai and Y. Chen, “Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes”, IEEE Transactions on Dependable and Secure Computing, Vol. 4, No. 1, January-March, 2007, pp.41-55.

[35] J. Balthrop, S. Forrest and M. R. Glickman, Revisiting “LISYS: Parameters and Normal Behavior”, Proceedings of the 2002 Congress on Evolutionary Computation, CEC, 2002, pp.1045-1050.

[36] Chung-Ming Ou, “Host-based intrusion detection systems adapted from agent-based artificial immune systems”, Neurocomputing, vol 88, 2012, pp78–86.

[37] P. Varalakshmi and S. Tharmarai Selvi, “Thwarting DDoS attacks in grid using information divergence”, Future Generation of Computer Systems, 2013, pp429-441.

[38] Chandrashekhar Azad, Vijay Kumar Jha, “Data Mining in Intrusion Detection: A Comparative Study of Methods, Types and Data Sets”, I.J. Information Technology and Computer Science, 2013, pp.75-90.

[39] Muhammad Aamir, Muhammad Arif, “Study and Performance Evaluation on Recent DDoS Trends of Attack & Defense”, I.J. Information Technology and Computer Science, 2013, 08, pp.54-65.

[40] Sutapa Sarkar, Brindha.M, “High Performance Network Security Using NIDS Approach”, I.J. Information Technology and Computer Science, 2014, 07, pp. 47-55.

Authors' Profiles

Dr. Inderpreet Chopra received his Bachelor's Degree with honors from Kurukshetra University in 2004, and obtained Master's Degree in Software Engineering from Thapar University in year 2006. He has also completed his PhD in Autonomic Grid Computing from Thapar University. He is presently working as a Manager with Expicient Inc. and has 9 years of experience in IT industry with expertise in design, development, and implementation. His area of interest includes Network Security, Distributed Computing, Grid Computing, Cloud Computing, SCM performance practices and Java/J2EE technologies.

Ratinder Kaur is a PhD scholar at Thapar University carrying out her research in the field of Network Security. She holds strong academic record. She received her Bachelor's Degree from Punjab Technical University and holds a Master's Degree, with honors in Software Engineering from Thapar University. She showcases strong inclination towards Computer Security field which is evident from her master thesis on Operating System fingerprinting, for which she won TCS (Tata Consultancy Services) Best Student Project Award, and now exploring Zero-day attack frontiers.

How to cite this paper: Inderpreet Chopra, Ratinder Kaur,"Immune-Inspired Self-Protection Model for Securing Grid", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.3, pp.75-84, 2016. DOI: 10.5815/ijitcs.2016.03.09