Secure User Tasks Distribution in Grid Systems

Maxim Kalinin
St. Petersburg State Polytechnical University
St. Petersburg, Russia
maxim.kalinin@ibks.ftk.spbstu.ru

Artem Konoplev
St. Petersburg State Polytechnical University
St. Petersburg, Russia
artem.konoplev@ibks.ftk.spbstu.ru

Dmitry Moskvin
St. Petersburg State Polytechnical University
St. Petersburg, Russia
dmitry.moskvin@ibks.ftk.spbstu.ru

Alexander Pechenkin
St. Petersburg State Polytechnical University
St. Petersburg, Russia
alexander.pechenkin@ibks.ftk.spbstu.ru

Dmitry Zegzhda
St. Petersburg State Polytechnical University
St. Petersburg, Russia
dmitry.zegzhda@ibks.ftk.spbstu.ru

Abstract—**The paper discusses a new approach to provide user tasks distribution in Grid systems using Petri nets unfolding. Branched Petri nets definition is proposed to describe Grid system security. Partial order method is applied to reduce size of Petri net model of Grid. Secure user tasks distribution method and system are suggested for automated security maintenance in the Grid. Solution of state explosion problem in branched Petri net model of Grid is proposed. Implemented access control system and obtained experimental results demonstrate successful solution of data protection against insiders in Grid systems.**

Keywords-grid; information security; Petri net; unfoldings; tasks distribution

I. INTRODUCTION

Implementation of security features in modern distributed computing systems, especially in Grid systems, which process confidential and restricted data, is accompanied by reduction of their scalability and parallelizability. It leads to preventing Grid systems from resource sharing thus turning it into a set of weakly bounded hosts.

Growing number of security violations relative to Grid systems (e.g., CVE-2009-0046 incident in Sun GridEngine, CVE-2013-4039 in IBM WebSphere Extended Deployment Compute Grid) proves high importance of task aimed at protection of data being processed in the Grid with minimal loss of Grid functionality.

Further, in Section 2 we consider existing works dedicated to solve this problem. Application of branching Petri nets to the Grid representation is provided in Section 3. Section 4 discusses the suggested solution of the state explosion problem basing on partial order method. In Section 5, we propose secure user tasks distribution method and system. Section 6 reviews the work results.

II. RELATED WORK

Grid systems provide the availability of increased amounts of valuable computing and information resources. Such information systems heavily depend on the provision of high security level. Naqvi and Riguidel [1] present a survey of the various Grid system threat models.

Special hardware and software components are used to provide protection against denial of service attacks and the spread of malicious software in Grid systems. These components also called security managers include Intrusion Detection Systems (IDS), firewalls and antivirus agents [2]. There are also several works aimed at solving the problem of anomaly detection in distributed computing systems [3][4].

Security managers are integrated with dedicated communication channels, which in the case of intrusion detection alerts are broadcast. After receiving such a notification, each host duplicates it to all resource providers being connected to it. As a result, all hosts isolate the problematic host. It thus prevents the possibility of attacks spreading in Grid systems.

In addition, in some Grid systems, fuzzy trust logic is used [5]. Each host is initially labeled. This label shows the trust level assigned to it by other components of Grid system. If attack from that host is fixed, the trust level is decreased. While search for a suitable host for a user-defined task, the hosts with the highest trust level are chosen for running this task.

In existing products, such as Grid Resource Allocation and Management (GRAM) in Globus Toolkit [6] and Community Authorization Service (CAS) in gLite [7], there are authentication and authorization mechanisms implemented to control user tasks access to Grid resources.

Definition and realization of security policies in these solutions are based on a set of Virtual Organizations (VOs) and fixed states of the Grid [8]. They do not take into account high dynamics and access rights distribution at the level of user tasks. Therefore, it might cause unauthorized access to data being processed in the Grid.

This paper refers to development of Grid system model, taking into account high dynamics of user tasks distribution, and suggests secure user tasks distribution method based on this model. In [9], an algorithmic behavior model of multi-agent distributed system is proposed. This model is based on adaptive random graphs mathematical apparatus and takes into account sufficiently high frequency of the number of nodes and links between nodes changing. There is high frequency of node status and user tasks distribution between nodes in Grid systems that can be observed. Whereas to add or delete a node in the Grid, you must pass the verification procedure which means that the number of nodes in such distributed network changes quite rarely.

In [10][11], an unfolding technique is formally described and applied to colored Petri nets to describe branching processes whose behavior close to the Grid. Branching

 ISBN: 978-1-61208-330-8

process is a Markov process [10] that models a population in which each individual in generation n produces some random number of individuals in generation $n + 1$. They propose a model of branching processes, suitable for describing the behavior of general Petri nets, without any finiteness or safeness assumption [11]. In this paper we extend the results of this work with reference to Grid systems security feature.

III. GRID SYSTEMS MODELING

Implementation of this approach involves the mathematical apparatus of functional colored Petri nets. A Grid system is described with Petri net $N = (RP, T, F, M)$, where $RP = \{rp\}$ is the finite set of vertices of graph which represents Grid nodes (hosts, resource providers, etc.), $T = \{t_i\}$ is a set of transitions between the vertices, $F = RP \times T \cup T \times RP$ is a set of arcs [12]. Markers $\{m\}$ denote user tasks from J (i.e., requests for a particular type of Grid resource).

T-transition of Petri net (N, M_0) or a simple transition is defined as a transition $t_{ij} \in T$, where mark M' directly accessible from mark M has the following form: $M' = (m_1, \ldots, m_i - 1, \ldots, m_j + 1, \ldots, m_n)$. Graphical model of T-transition is presented in Fig. 1.

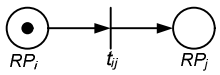

Figure 1. T-transition.

F-transition of Petri net (N, M_0) or branching is defined as a transition $t_{ijk} \in T$, where mark M' directly accessible from marking M has the following form: $M' = (m_1, \ldots, m_i - 1, \ldots, m_j + 1, \ldots, m_k + 1, \ldots, m_n)$. Graph representation of F-transition is provided in Fig. 2.

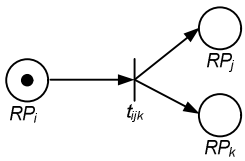

Figure 2. The graph representation of F-transition.

Let us define the branching Petri net as a subclass of colored functional Petri net.

Definition 1. The branching Petri net is a colored functional unlimited Petri net which has only T- and F-transitions.

Any Grid system can be represented as a graph of branching Petri net. T-transition means the simple migration of user request from one node to another. There are no new markers appear in that case, which means that the summary marker counts taken before and after transition activation are the same.

F-transition means situation when the computing power of multiple Grid nodes is required to cope with user task. The total count of markers is increased according to the number of nodes involved into user task processing.

Any user task migration in the Grid keeps arcs multiplicity at the level of one. Taking into account specified definition, there are only T- and F-transitions can exist in the Grid.

IV. SOLUTION OF STATE EXPLOSION PROBLEM

While modeling such a high distributed systems as Grid systems number of states grows exponentially with an increase in the number of nodes. For example, an initial marking of any Petri net representing the Grid has the following form: $M_0 = (m_1, \ldots, m_n)$. Each marker of this marking set assumes a value in the range of 0 to n_A, where n_A is a number of active nodes of Grid. Assume that there is no user task can be produced on any node before previous one would have been finished ($n_A \leq n$). Then the total number of states describing such Grid is n_A^n (e.g., $n = 1000$, then power of states set is 10^{3000}). These problem is known as a state explosion problem.

There is a partial order method implemented to solve this problem. Define the partial order on the set of markings of Petri net. Let M_1 and M_2 be the markings of Petri net (N, M_0). Assume that $M_1 \leq M_2$ being in a partial order relationship, if for every marker m of marking M_1 situated in p position, there is a marker m' of marking M_2 in the same position p, where m' is equal to or greater than m.

M_1 is less than M_2 relatively to order $\leq$, if marking M_1 can be obtained from marking M_2 by sequential markers removing from Petri net vertices. Using this fact as a basis, an inductive definition of minimal partial order can be done.

Definition 2. Minimal partial order of marking M of Petri net is a natural number D such that for any marking M' being in partial order with M: $M' \leq M$, there is no marker m' of marking M', where $m' < D$.

Lemma 1. Minimal partial order of the branching Petri net marking is equal to 1.

Proof. According to definition 1, the branching Petri net is unlimited. It means that it could be any number of markers (user tasks) at any position. At least one marker at the position is enough for transition to be triggered because of the multiplicity of branched Petri net. Therefore, 1 is a minimal natural number to which it is possible to decrease the amount of markers at every position of Petri net.

Theorem 1. Any marking of the branched Petri net N reachable from marking M is also reachable from marking $M' = (m'_1, \ldots, m'_n)$, $m_i' = \{0, 1\}$ which is obtained from N by applying to it the partial order equal to 1.

Proof. Consider the simple case when the branched Petri net is 2-limited. Common case can be proved in induction. For every $m \in M : m \leq 2$. By the definition the branching Petri net consists of aggregate of T- and F-transitions. Consecutively, consider all possible ways of such Petri net fragment aggregation.

 ISBN: 978-1-61208-330-8

- T-transition—T-transition. Branched Petri net fragment of such kind has 1 of 2 forms, as shown in Fig. 3.

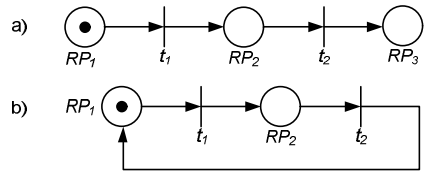

Figure 3. The branched Petri net fragment of T-transition—T-transition type.

In that case, the amount of markers at each position can be arbitrarily large. Find the set of reachable states for each form of branched Petri net fragment, taking into account deterministic form of transition function which means that *ceteris paribus* transitions are triggered simultaneously. For simplicity we also agree that all markers have the same type.

The fragment of Petri net illustrated in Fig. 3 generally has marking M_1 = (a, b, c), where a, b, c — natural numbers, and has the following set of the states: $R(M_1)$ = {(k, l, m), (0, l, c), (k, 0, m), (k, l, 0), (k, 0, 0), (0, l, 0), (0, 0, m), (0, 0, 0)}, where k, l, m — natural numbers and max(k, l, m) ≤ max(a, b, c).

Apply net partial order relationship, where D = max(a, b, c) - 1. Then the resulting Petri net has marking M_1' = (x, y, z), where x, y, z — natural numbers and max(x, y, z) = max(a, b, c) – 1. Therefore $R(M_1')$ = {(k', l', m'), (0, l', c'), (k', 0, m'), (k', l, 0), (k', 0, 0), (0, l, 0), (0, 0, m'), (0, 0, 0)}, where k', l', m' — natural numbers and max(k', l', m') ≤ max(x, y, z) ≤ max(a, b, c), i.e., $R(M_1) = R(M_1')$.

Thereby in induction: $R(M_1) = R(M_1')$ for $\forall D \in N : M_1' \leq M_1$. According to the Lemma 1 the minimal partial order of branching Petri net marking $D_{min} = 1$. In addition $\forall m' \in M' : m' = \{0, 1\}$. Thus, we have $M' = (m'_1, \ldots, m'_n)$, $m_i' = \{0, 1\}$.

- F-transition—F-transition. Branched Petri net fragment of such kind has 1 of 2 forms, as shown in Fig. 4.

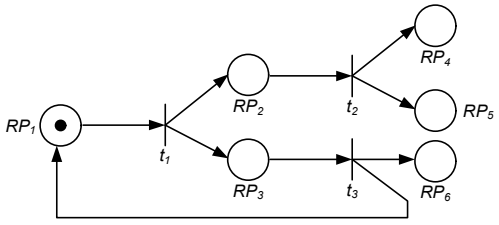

Figure 4. The branched Petri net fragment of F-transition—F-transition type.

The fragment of Petri net illustrated in Fig. 4, generally, has marking M_2 = (a, b, c, d, e, f), where a, b, c, d, e, f — natural numbers and has the following set of reachable states $R(M_2)$ = {(k, l, m, n, o, p), (0, 0, 0, n, o, 0), (0, 0, 0, n, 0, 0), (0, 0, 0, 0, o, 0), (0, 0, 0, 0, 0, p), (0, 0, 0, 0, 0, 0)}, where k, l, m, n, o, p — natural numbers and min(k, l, m) ≥ min(a, b, c, d, e, f).

Apply partial order relationship, where D = min(a, b, c, d, e, f) + 1. Then, the resulting Petri net has marking M_2' = (x, y, z, α, β, χ), where x, y, z, α, β, χ — natural numbers and min(x, y, z, α, β, χ) = min(a, b, c, d, e, f) + 1. Hence $R(M_2')$ = {(k', l', m', n', o', p'), (0, 0, 0, n', o', 0), (0, 0, 0, n', 0, 0), (0, 0, 0, 0, o', 0), (0, 0, 0, 0, 0, p'), (0, 0, 0, 0, 0, 0)}, where k', l', m', n', o', p' — natural numbers and min(k', l', m', n', o', p') ≥ min(x, y, z, α, β, χ) ≥ min(a, b, c, d, e, f), i.e., $R(M_2) = R(M_2')$.

Thereby in induction: $R(M_2) = R(M_2')$ for $\forall D \in N : M_2' \leq M_2$. According to the Lemma 1 the minimal partial order of branching Petri net marking $D_{min} = 1$. Thus, we have $\forall m' \in M' : M' = (m'_1, \ldots, m'_n)$, $m_i' = \{0, 1\}$.

- F-transition—T-transition. Branched Petri net fragment of such kind has 1 of 2 forms, as shown in Fig. 5.

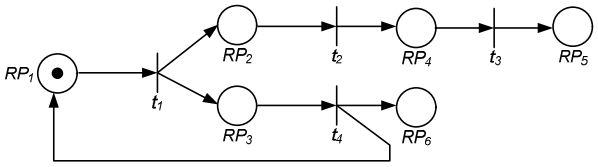

Figure 5. The branched Petri net fragment of F-transition—T-transition type.

The fragment of Petri net illustrated in Fig. 5 is a composition of fragments illustrated in Figs. 3-4. Following the induction, we have $R(M_3) = R(M_3')$ for $\forall D \in N : M_3' \leq M_3$, where M_3 — marking of specified Petri net fragment. Thus, we get $\forall m' \in M' : M' = (m'_1, \ldots, m'_n)$, $m_i' = \{0, 1\}$.

The provisions of this theorem allow to reduce a set of the states which describe the Grid from to n_A^n to 2^n.

V. SECURE USER TASKS DISTRIBUTION

Current security-based solutions referenced to describing and enforcement of security policies operate with a set of virtual organizations and fixed Grid states. These solutions do not take into account real access rights distribution at the level of user tasks running on Grid system nodes. They also miss the fact of high intensity of user tasks migration between such nodes. Thus, it leads to the possibility of unauthorized access to processed data.

The proposed method of secure user tasks distribution is based on the solution of a reachability problem in terms of Petri net describing the Grid. There is a reachability graph suggested to create for the specified Petri net N= (RP, T, F, M) with initial marking $M_0 = (m_1, \ldots, m_n)$, where vertices are the marking with minimum partial order equal to 1. According to the Theorem 1, that tree must be a finite one. Vertices of this graph form a set of states which the system may reach. For every state, a formalized transition function

is used to determine compliance with security policy constraints. Verification is performed by comparison of security policy requirements with the current state.

As a result, there is the set of Grid nodes user tasks, transmission to which is permitted by security policy rules. After that, the rules of user tasks distribution are transmitted to such nodes (Fig. 6).

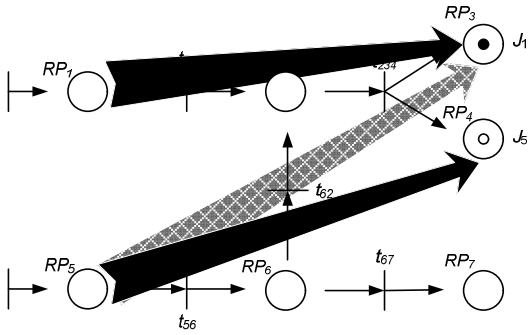

Figure 6. Secure user tasks distribution scheme

The secure user tasks distribution system is proposed and its effectiveness estimation shown. Total time required to process some user task in the Grid includes a time of user task processing by n nodes, time of data transmission between n nodes and time required to define permitted task distribution according to the discussed method.

$$T = t_p + t_m + t_s, \quad (1)(1)$$

According to (1), t_p is a time of user task processing by n nodes; t_m is a time of data transmission between n nodes; t_s is a time required to define permitted task distribution according to the discussed method.

$$t_p = \frac{k_1}{s(n)}, s(n) = \frac{1}{k_2 + \frac{1-k_2}{n}}, \quad (2)$$

According to (2), k_1 is a total time of user task processing by single node; k_2 is a portion of the task that cannot be parallelized (it remains serial).

$$t_m = n\frac{k_3}{k_4}, t_s = \frac{k_5 \cdot 2^n}{k_6}, \quad (3)$$

According to (3), k_3 is an amount of data required for user task processing being transported to each Grid node; k_4 is a data rate; k_5 is a number of operations required for Grid state verification that is being represented by the reachability tree of branched Petri net; k_6 is a computing power of node.

Relative reduction in time costs required for processing of restricted information in Grid system is calculated as a division of time required to process user task with proposed method and without it.

$$Q = \frac{\sum_{i=1}^{d}(k_{1i} \cdot (k_2 + \frac{(1-k_2)\cdot c}{n}) + \frac{n}{c} \cdot \frac{k_3}{k_{4i}})}{\sum_{i=1}^{d}(k_{1i} \cdot (k_2 + \frac{(1-k_2)}{n}) + n \cdot \frac{k_3}{k_{4i}} + \frac{k_5 \cdot 2^n}{k_{6i}})}, \quad (4)$$

In (4), c is a number of restricted data classification categories being processed in the Grid; d is a number of iterations used to perform the specified task.

Typical user tasks in complex distributed analytical systems which use Grid software to perform processing of huge amount of data (e.g., CERN [8]) have the following parameters: $k_1 \approx 1$ hour, $k_2 \approx 20\%$, $k_3 \approx 512$KB, $k_4 \approx 100$ Mb/sec, $k_5 \approx 102$ oper, $k_6 \approx 4 * 1010$ oper/sec, $n \approx 2000$, $c = 5$ (sample). Experimental results for Grid sample with such characteristics are shown in Fig. 7. As one can see, Grid system with integrated proposed access security subsystem requires significantly less time to process user tasks than Grid system with organizational measures aimed to divide the Grid in isolated segments to prevent restricted data leaks. Experiments have been performed on a laboratory stand included of 300 compute nodes, which are virtual machines of multiprocessor computer running on Xen Cloud Platform [13]. Software infrastructure is based on Globus Toolkit 5 [6].

Implementation of secure user tasks distribution system in the Grid allows us to protect data from user privilege escalation, simultaneously reduce the time expenses associated with the security assessment and increase the productivity of Grids which processes classified data.

VI. CONCLUSION AND FUTURE WORK

The new approach of Grid systems modeling based on mathematical apparatus of Petri nets is proposed. Subclass of colored Petri nets called 'branched Petri nets' is used to represent behavior of the Grid. Extremely large size of models representing real high distributed systems causes problem known as state explosion. Partial order method is applied to branched Petri net to solve it.

Proposed secure user tasks distribution method based on technique of reachability tree construction and subsequent security verification of obtained tree nodes. Implemented access control system provides successful solution of data protection against attacks based on user privilege escalation technique.

Future work of proposed solution involves access control system integration to most popular Grid software infrastructures. Different types of authorization techniques also must be taken into account.

REFERENCES

[1] S. Naqvi and M. Riguidel, "Threat model for grid security services", Lecture Notes in Computer Science, vol. 3470, 2005, pp. 1048-1055, doi:10.1007/11508380_107.

 ISBN: 978-1-61208-330-8

[2] H. Lohr, H. V. Ramasamy, A. Sadeghi, S. Schulz, M. Schunter, and C. Stuble, “Enhancing grid security using trusted virtualization”, Lecture Notes in Computer Science, vol. 4610, 2007, pp. 372-384, doi: 10.1007/978-3-540-73547-2_39.

[3] T. Stepanova, D. Zegzhda, M. Kalinin, and P. Baranov, “Mobile anomaly detector module based on power consumption analysis”, Proc. International Conference on Information Security and Privacy (ISP-10), Jul. 2010, pp. 85-89.

[4] M. Burgess, “Probabilistic anomaly detection in distributed computer networks”, Science of Computer Programming, vol. 60, Mar. 2006, pp. 1-26, doi:10.1016/j.scico.2005.06.001.

[5] S. Song, K. Hwang and M. Macwan, “Fuzzy trust integration for security enforcement in grid computing”, Lecture Notes in Computer Science, vol. 3222, 2004, pp. 9-21, doi: 10.1007/978-3-540-30141-7_6.

[6] D. Gomez, “Secure collaborative grid computing”, Johns Hopkins Whiting School of Engineering, Spring 2008.

[7] L. Pearlman, C. Kesselman, V. Welch, I. Foster, and S. Tuecke, “The community authorization service: status and future”, Proc. of Computing in High Energy Physics 03 (CHEP '03), 2003, pp. 44-52.

[8] A. Chakrabarti, “Grid computing security”, Berlin: Springer, 2007.

[9] T. Stepanova and D. Zegzhda, “Stochastic model of interaction between botnets and distributed computer defense systems”, Computer Network Security, Lecture Notes in Computer Science, vol. 7531, 2012, pp. 218-225, doi: 10.1007/978-3-642-33704-8_19.

[10] V. Kozura, “Unfoldings of coloured Petri nets”, Lecture Notes in Computer Science, vol. 2244, 2001, pp. 268-278, doi:10.1007/3-540-45575-2_27.

[11] J. Couvreur, D. Poitrenaud and P. Weil, “Branching processes of general Petri nets”, Lecture Notes in Computer Science, vol. 6709, 2011, pp. 129-148, doi: 10.1007/978-3-642-21834-7_8.

[12] P. Zegzhda, D. Zegzhda, M. Kalinin, and A. Konoplev, “Security modeling of grid systems using Petri nets”, Computer Network Security, Lecture Notes in Computer Science, vol. 7531, 2012, pp. 299-308, doi: 10.1007/978-3-642-33704-8_25.

[13] “XCP Overview”, web site: *wiki.xenproject.org/wiki/XCP_Overview*.

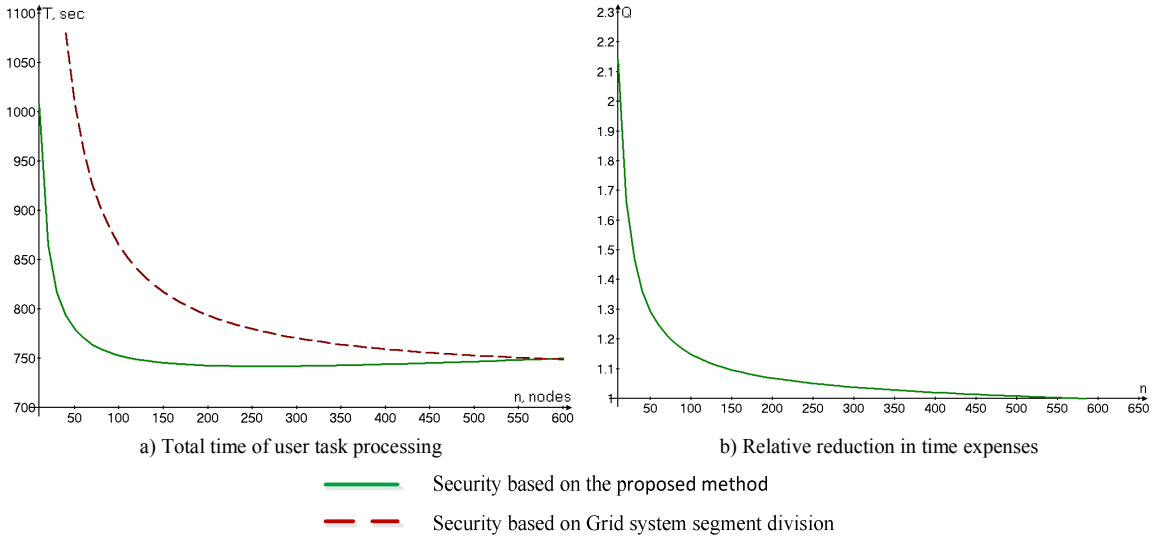

Figure 7. Secure user tasks distribution system experimental results.