ОБНАРУЖЕНИЕ ВИРУСНЫХ АТАК В РАСПРЕДЕЛЕННЫХ СРЕДАХ И ОДНОРОДНЫХ СЕТЯХ КРИТИЧЕСКОГО ПРИМЕНЕНИЯ

К.С. Ткаченко, Н.Л. Корепанова

Севастопольский национальный технический университет
г. Севастополь, ул. Университетская, 33
E-mail: tkachenkokirillstanislavovich {@mail.ru, @gmail.com}

В статье предлагается метод обнаружение вирусных атак в распределенных средах и однородных сетях критического применения на основе критерия для параметров сетей при априорной неопределенности входных данных.

Введение. Проблема построения гарантопригодных систем критического применения важна, нова и актуальна, поскольку связана с важными научными и практическими задачами построения однородных вычислительных систем в условиях вирусных атак (ВА).

В работе [1] описывается внутренняя архитектура *GRID*-системы. В публикации [2] предлагается способ оценки качества обслуживания абонентов. Книга [3] содержит описание ряда алгоритмов, связанных с расчетами для марковских цепей и сетей массового обслуживания. В статье [4] представляется алгоритм Бузена. В информационном документе [5] воссоздается стратегия для решения вопросов безопасности. В проектном предложении [6] поднимаются вопросы, связанные с требованиями к безопасности. В рекомендациях [7] излагаются правила для базового уровня безопасности.

Нерешенной прежде является задача обнаружения ВА на основе параметров сетей.

Целью данной работы является разработка метода обнаружения ВА в распределенных средах и однородных сетях критического применения.

Одной из многих используемых на практике конфигураций *GRID*-систем (в том числе и для критического применения) и балансировщика нагрузки является *GridWay* [1] (далее – сеть). При достаточно сильном функциональном укрупнении любую подобную сеть S_G можно представить в виде

$$S_G = \bigcup_{i=0}^{10} S_i . \qquad (1)$$

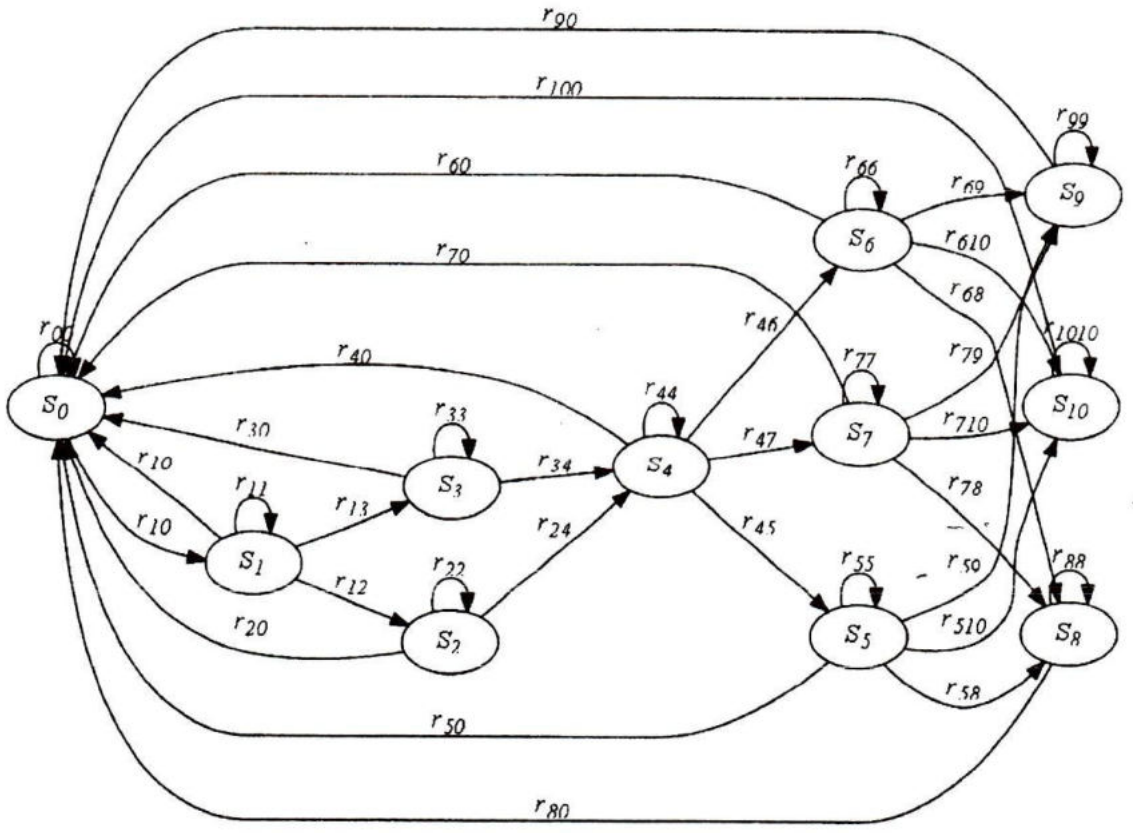
При этом:

S_0 – пользовательский интерфейс. Предоставляет конечному пользователю набор команд для подачи, уничтожения, переноса, мониторинга и синхронизации рабочих пакетов заданий. Включает в себя программный интерфейс управления распределенными ресурсами приложений на языках программирования высокого уровня и возможности командной строки; S_1 – менеджер запросов; S_2 – пул пакетов запросов; S_3 – пул хостов; S_4 – диспетчер для выполнения предоставления и мониторинга выполнения пакетов заданий; S_5 – информационный менеджер для мониторинга хостов; S_6 – менеджер выполнения для управления выполнением пакетов заданий; S_7 – менеджер передачи для удаленной настройки хостов; S_8 – службы передачи файлов; S_9 – службы выполнения; S_{10} – информационные службы.

Процессы, протекающие в сети, происходят в дискретные моменты на непрерывном множестве значений времени; однородны, поскольку не зависят от сдвигов по оси времени; из любого состояния можно перейти в любое за конечное число тактов. Поэтому считая, что протекающий процесс можно описать неприводимой дискретной однородной марковской цепью, в которой состояния S_i, $i = \overline{0,10}$ из (1), и на основании рисунка «*Components of the GridWay Meta-scheduler*» [1] строится матрица передач R:

$$R = \begin{pmatrix} r_{00} & r_{01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{10} & r_{11} & r_{12} & r_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{20} & 0 & r_{22} & 0 & r_{24} & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{30} & 0 & 0 & r_{33} & r_{34} & 0 & 0 & 0 & 0 & 0 & 0 \\ r_{40} & 0 & 0 & 0 & r_{44} & r_{45} & r_{46} & r_{47} & 0 & 0 & 0 \\ r_{50} & 0 & 0 & 0 & 0 & r_{55} & 0 & 0 & r_{58} & r_{59} & r_{510} \\ r_{60} & 0 & 0 & 0 & 0 & 0 & r_{66} & 0 & r_{68} & r_{69} & r_{610} \\ r_{70} & 0 & 0 & 0 & 0 & 0 & 0 & r_{77} & r_{78} & r_{79} & r_{710} \\ r_{80} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{88} & 0 & 0 \\ r_{90} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{99} & 0 \\ r_{100} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_{1010} \end{pmatrix}. \quad (2)$$

В (2) r_{ij} – предполагаемая вероятность перехода из состояния S_i в состояние S_j. Все оценки значений r_{ij} можно получить статистически, на основании расширенного анализа протоколов функционирования конкретного экземпляра сети. Предполагается, что все потоки – простейшие. На рис. 1 изображается граф переходов цепи.



Р и с. 1. Граф переходов цепи

Система линейных алгебраических уравнений, описывающих трафик, строится на основании (2) как [2]

$$\begin{aligned} &\nu_1 = 1, \\ &(v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}) = \\ &= (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}) \cdot \\ &\cdot R \end{aligned} \quad (3)$$

В (3) ν_i – суммарная интенсивность на выходе i-й подсистемы.

Получив из (3) ν_i $\left(i = \overline{0,10}\right)$, можно на основании итеративных формул алгоритма Бузена [3, 4] получить вектор нормализующих констант $g(i)$ $\left(i = \overline{0,10}\right)$ путем расчета по алгоритму:

Шаг 1. $g(0) := 1$; $\forall i = \overline{1,10}$ Выполнять $g(i) := 0$;

Шаг 2. $\forall j = \overline{1,10}$ Выполнять ($\forall i = \overline{1,10}$ Выполнять $g(i) := g(i) + \nu_j \cdot g(i-1)$).

Алгоритм в тождественных обозначениях может быть записан на Java как

```
g[0] = 1;
for (int i = 1; i < g.length; i++) {
  g[i] = 0;
}
for (int j = 1; j < g.length; j++) {
  for (int i = 1; i < g.length; i++) {
    g[i] += v[j] * g[i - 1];
  }
}
```

Поскольку на основании $g(i)$ $\left(i = \overline{0,10}\right)$ возможно получить стационарные значения вероятностей пребывания в любом состоянии [3] S_i, $i = \overline{0,10}$, то будем оценивать факт наличия вирусной атаки (ВА) на основе сравнения значения $g(0)_I$ в исследуемой системе с некоторыми образцовыми значениями $g(0)$ в системах с ВА и без ВА. Пусть имеется n_{BA} образцов $g(0)$ для систем под воздействием ВА и $n_{\overline{BA}}$ образцов $g(0)$ без ВА. На основании простого критерия знаков последовательно проверяется гипотеза H_0 ={нет различия в параметре положения, нет сдвига}: I. для систем под воздействием ВА; II для систем без ВА. Для ситуаций I и II рассчитывается число нулевых сдвигов n сравнительно с $g(0)_I$, число положительных сдвигов B^+, число отрицательных сдвигов B^-. На их основании рассчитывается значимость Z:

$$Z = 0{,}5^n \sum_{k=\max\{B^-, B^+\}}^{n} C_n^k + 0{,}5^n \sum_{k=0}^{\min\{B^-, B^+\}} C_n^k \quad (4)$$

При этом расчеты по формуле (3) удобно выполнять с использованием САПР Maxima и Octave, формуле (4) – офисных средств Gnumerics и Apache OpenOffice/LibreOffice Calc, по алгоритму Бузена – с использованием программной системы (ПС), написанной на языке программирования Java или подобном императивном языке программирования высокого уровня.

Могут возникнуть информационные ситуации, перечисленные в таблице 1.

Таблица 1.

Информационные ситуации

№	A	B
1.	–	–
2.	–	+
3.	+	–
4.	+	+

В табл. 1 № – номер информационной ситуации, A – значимость при сравнении с образцами для систем с ВА, B – значимость при сравнении с образцами для систем без ВА, «+» – значимость велика, «–» – значимость мала. При информационной ситуации № 3 считается, что в исследуемой системе наблюдается ВА, в прочих ситуациях – ВА отсутствует. В случае предполагаемого обнаружения ВА необходимо принять ряд мер системным и сетевым администраторам в директивном порядке. Поскольку при ВА может значительно возрастать загрузка сети, необходимо произвести увеличение производительности отдельных подсистем, а именно S_1, S_2, S_3.

Для этих целей рекомендуется компьютеры, на которых функционируют S_1, S_2, S_3, настроить на принудительную работу с ограничением полезной производительности. Затем в S_0, поскольку перед S_0 возможно имеется узкая полоса пропускания, необходимо сбросить множественные соединения на один порт, закрыть порты для *ICMP*-запросов и увеличить одновременное количество максимальных подключений к базе данных сервера, установить производительный кэширующий сервер непосредственно перед *Web*-сервером. Чтобы в условиях ВА происходило эффективное и корректное взаимодействие между S_i [5 – 7], необходимо на уровне политики безопасности внести ряд изменений. В случае если имеется поток от конечного пользователя, запрашивающего целевую услугу, и запрос проходит через посредников, подключается многоступенчатая система аутентификации. Пользователь может получить удостоверение по аутентификации на локальном сервере аутентификации и предоставить учетные данные как часть запроса на обслуживание. В ситуации когда получаемый запрос перенаправляется через шлюз, шлюз может получить определенные атрибуты и привилегии пользователя и отправлять запросы с ними. Следует помнить о том, что инфраструктура S_{ci} [6] не может быть быстро изменена, и, следовательно, описываемые архитектура и требования безопасности должны интегрироваться с существующей инфраструктурой безопасности заранее.

Заключение. В данной работе приведен метод обнаружение ВА в распределенных средах и однородных сетях критического применения на основе критерия для параметров сетей при априорной неопределенности входных данных. Перспективой дальнейших изысканий по данной тематике станет применение наработок при разработке инструментального средства для контроля и оперативного управления распределенной средой.

СПИСОК ЛИТЕРАТУРЫ

1. *GridWay* Internal Architecture [Электронный ресурс] / Ресурс переменной длины. – Режим доступа: http://gridway.org/doku.php?id=documentation:release_5.14:ia 138094 bytes 07.07.2014 12.41, свободный. – Загл. с экрана. (Online).
2. *Жабрев В.С.* Модель оценки качества обслуживания абонентов в виде системы массового обслуживания / В.С. Жабрев, В.В. Прокопенко // Информационно-управляющие системы, № 3, 2008. – Южно-Уральский государственный университет. – 2008. – С. 23 – 26.
3. *Stewart W.J.* Probability, Markov chains, queues and simulation: the mathematical basis of performance modeling / William J. Stewart. – Woodstock: Princeton University Press, 2009. – 758 с.
4. *Buzen's* algorithm [Электронный ресурс] / Ресурс переменной длины. – Режим доступа: http://en.wikipedia.org/w/index.php?title=Buzen%27s_algorithm&printable=yes 114553 bytes 28.08.2014 20.07, свободный. – Загл. с экрана. (Online).
5. *Security* Architecture for Open Grid Services [Электронный ресурс] / Ресурс переменной длины. – Режим доступа: http://toolkit.globus.org/toolkit/security/ogsa/draft-ggf-ogsa-sec-arch-01.pdf 456802 bytes 31.08.2014 12.00, свободный. – Загл. с экрана. (Online).
6. *CS590L* Distributed Component Architecture [Электронный ресурс] / Ресурс переменной длины. – http://www.manishmehta.com/academics/CS590L/CS590L%20Project%20Proposal.pdf 31134 bytes 31.08.2014 12.00, свободный. – Загл. с экрана. (Online).
7. *OGSA®* Basic Security Profile 2.0 [Электронный ресурс] / Ресурс переменной длины. – http://www.ogf.org/documents/GFD.138.pdf 87749 bytes 31.08.2014 12.00, свободный. – Загл. с экрана. (Online).