А. А. Станкевичус
Московский инженерно-физический институт (государственный университет)

О НЕКОТОРЫХ МЕТОДАХ ЗАЩИТЫ ГРИД ОТ ВРЕДОНОСНОГО КОДА

В настоящей работе проводится обзор методов защиты Грид от вредоносного кода, их преимуществ и недостатков, рассматривается возможность создания нового метода защиты, сочетающего в себе преимущества приведенных методов.

Введение

С развитием науки и техники появилось множество прикладных и научных задач, требующих вычислительных мощностей, недоступных в рамках отдельных компьютерных центров. Возникла идея объединения ресурсов множества ЭВМ для решения ресурсоемких задач, в результате развития которой в 90-х годах XX в. начала складываться технология Грид. В числе первых технологиями Грид воспользовались физики из Европейского центра ядерных исследований — ЦЕРН [1]. Там Грид был необходим для обработки данных, поступающих от крупнейшего в мире ускорителя заряженных частиц — Большого адронного коллайдера (Large Hadron Collider, LHC [2]). В настоящее время Грид используются для решения множества задач по самым разным фундаментальным и прикладным направлениям — в физике высоких энергий и космофизике, микробиологии и медицине, метеорологии, самолетостроении и целом ряде других областей [3, 4, 5].

Грид — это географически распределенная инфраструктура, в ее состав входит множество разнотипных ресурсов (таких как запоминающие устройства и процессоры, базы данных и сети), доступ к которым обеспечивается пользователю независимо от его месторасположения и этих ресурсов. Одно из определений этого термина дано Я. Фостером и К. Кессельманом: «Грид (Grid) — согласованная, открытая и стандартизованная среда, которая обеспечивает гибкое, безопасное, скоординированное разделение ресурсов в рамках виртуальной организации» [6]. «Грид предполагает коллективный разделяемый режим доступа к ресурсам и к связанным с ними услугам в рамках глобально распределенных виртуальных организаций, состоящих из предприятий и отдельных специалистов, совместно использующих общие ресурсы. В каждой виртуальной организации имеется своя собственная политика поведения ее участников, которые должны соблюдать установленные правила. Виртуальная организация может образовываться динамически и иметь ограниченное время существования» [7].

Пользователю Грид может предоставляться доступ к большому количеству различных ресурсов, причем каждый из этих ресурсов может использоваться большим количеством пользователей, обрабатывающих различные данные. Так, например, в проекте World Community Grid насчитывается **402626 пользователей и 1070890 компьютеров [8].** В случае если доступ к Грид получит злоумышленник, его действия могут повлиять на работу большого числа ресурсов и пользователей Грид, при этом злоумышленник может нанести колоссальный ущерб и ресурсам и пользователям Грид.

В Грид можно выделить следующие действующие лица: пользователь прикладного ПО Грид, производитель прикладного ПО Грид, владелец ресурса Грид, владелец диспетчера Грид, производитель промежуточного ПО Грид. При этом количество пользователей прикладного ПО Грид, производителей ПО Грид и владельцев ресурса Грид может быть очень велико, что затрудняет установление доверительных отношений между ними. Кроме того, существует реальная угроза распространения вредоносного кода пользователем прикладного ПО Грид и производителем прикладного ПО Грид. Проблемы безопасности Грид рассматриваются, например, в работах [9, 10, 11, 12]. Вопросы защиты Грид от вредоносного кода рассматриваются, в частности, авторами работ [13, 14]. Проблема безопасности Грид очень остра, вредоносный код может нарушать работу ресурсов Грид, а также Грид в целом, Грид может быть использована как платформа для реализации распределенных атак.

Вышесказанное определяет актуальность разработки методов защиты Грид от вредоносного кода. Задачей настоящей работы является рассмотрение и анализ методов защиты Грид от вредоносного кода, их преимуществ и недостатков, рассмотрение возможности создания нового метода защиты, сочетающего в себе преимущества приведенных методов.

1. Методы защиты Грид от вредоносного кода

На данный момент во многих системах Грид используется инфраструктура открытого ключа (ИОК) [15] для надежной аутентификации пользователей и производителей прикладного ПО Грид, что позволяет в случае распространения ими вредоносного кода использовать административные меры наказания. Однако такие методы не могут гарантировать безопасность, так как вредоносные действия могут быть обнаружены и прекращены уже после реализации угрозы. Для повышения уровня защищенности могут быть применены различные технические средства защиты от вредоносного кода.

Защита от вредоносного кода может быть реализована путем применения: технологий виртуализации [14], безопасных языков программирования [16], верификации исходного или исполнимого кода [17], верификации исходного кода [18].

Во многих используемых системах применяются именно технологии виртуализации. К сожалению, это, как правило, ведет к существенному падению производительности вычислительных ресурсов Грид либо налагает дополнительные требования к среде выполнения и аппаратному обеспечению этих ресурсов, кроме того, безопасность многих виртуальных машин не имеет строгого доказательства и уже существуют известные уязвимости некоторых из них [19].

Использование безопасных языков программирования [16] может позволить решить проблему безопасности, однако код на этих языках на данный момент может выполняться не на всех платформах, производительность этого кода во многих случаях уступает производительности кода на небезопасных языках, а сами безопасные языки зачастую не очень удобны для разработчиков и их использование может требовать дополнительных затрат на обучение разработчиков и разработку прикладного ПО.

Верификация [17] исполнимого кода во многих случаях затруднена в связи с необходимостью анализа исполнимого кода, который, в свою очередь, до сих пор является плохо поддающейся автоматизации задачей. Кроме того, верификация исполнимого кода для различных программных и аппаратных платформ, как правило, требует разработки различных средств для каждой из платформ.

Верификация [18] исходного кода может позволить гарантировать безопасность исходного кода, не предъявляя дополнительных требований к аппаратному обеспечению или среде исполнения, причем верификация может быть выполнена для программ на широко распространенных языках, позволяющих получать высокопроизводительный и переносимый код, таких, как язык Си.

Часто безопасность ПО обеспечивается путем доказательства его корректности или безопасности типов, из чего следует в том числе и безопасность этого ПО. Существует множество инструментов, предназначенных для доказательства корректности работы программ, однако эти инструменты, как правило, доказывают корректность не для всех программ, а лишь для программ, использующих некоторое подмножество языка. Важно понимать, что корректность не эквивалентна безопасности программы. Безопасная программа может быть в общем случае некорректной, например, она может допускать ситуации, в которых может происходить переполнение буфера, не влекущее никаких опасных последствий, а, скажем, ведущее всего лишь к получению неточного результата вычислений. Таким образом, для решения задачи защиты вычислительной среды от вредоносного кода не требуется доказательства корректности исполняемой программы и достаточно доказательства ее безопасности для этой среды.

Проблема корректности результатов вычисления, как правило, возникает в критически важных приложениях и касается разработчика программного обеспечения и его пользователей, но не влияет на безопасность вычислительного узла. Эта задача выходит за рамки области методов защиты Грид от вредоносного кода.

Безопасность типов также является в общем случае более жестким требованием, чем безопасность исходного кода, так как многие операции, нарушающие безопасность типов, не приводят к нарушению безопасности среды исполнения.

Таким образом, существует несколько основных подходов к защите Грид от вредоносного кода. Методы в рамках каждого из подходов имеют свои преимущества и недостатки. Рассмотрим возможности устранения перечисленных недостатков.

2. Доказательство безопасности вычислительного узла Грид

Разработка и использование узкоспециализированных методов и средств, ориентированных на обеспечение безопасности именно вычислительных ресурсов Грид, может позволить проводить верификацию и оптимизацию на уровне абстракции, соответствующем предметной области. В общем случае доказательство безопасности вычислительного ресурса Грид проще, чем доказательство безопасности произвольного ПО, доказательство корректности ПО или обеспечение безопасности типов.

Язык программирования Си является широко распространенным, существуют компиляторы этого языка для практически любой платформы, кроме того, производительность программ на языке Си во многих случаях превышает производительность аналогичных программ на других языках программирования, что делает этот язык удобным для разработки приложений для Грид. Однако этот язык не является типобезопасным и в общем случае для программ на этом языке корректность на данный момент доказать не удается, что часто является следствием высокой производительности. Таким образом, существует необходимость в доказательстве безопасности исходного кода прикладного ПО для Грид на языке Си.

Для реализации решения задачи защиты Грид-систем от вредоносного кода в данной работе предлагается использование автоматической верификации исходного кода прикладного ПО Грид, написанного на языке Си, в результате которой для многих программ на языке Си может быть доказана безопасность для вычислительной среды, причем многие программы, для которых безопасность не может быть доказана непосредственно, могут быть автоматически дополнены проверками или транслированы в эквивалентную форму, допускающую автоматическое доказательство их безопасности. При этом влияние дополнительных проверок, необходимых для обеспечения безопасности, на производительность программы может быть не столь значительно, как влияние использования типобезопасных языков или применения программ, для которых может быть автоматически доказана корректность.

Представляются перспективными следующие направления исследований:

1. Поиск путей повышения уровня защищенности ресурсов Грид от вредоносного кода на основе анализа состояния вопросов теории и практики защиты Грид, современных тенденций развития.

2. Систематизация способов защиты от вредоносного кода, оценка их эффективности.

3. Развитие теоретических положений по защите Грид от вредоносного программного обеспечения.

4. Поиск и разработка новых решений в области защиты Грид от вредоносного кода на основе известных теоретических положений и новых методов защиты.

Задачи разработки методики защиты Грид от вредоносного кода, а также разработки на основе этой методики инструментального средства, обеспечивающего возможность безопасного применения в Грид разработанного на языке Си прикладного программного обеспечения из недоверенных источников, выходят за рамки настоящей работы и могут быть рассмотрены в последующих работах.

Заключение

В настоящей работе были рассмотрены методы защиты Грид от вредоносного кода, их преимущества и недостатки. На основе анализа существующих методов были сделаны выводы о возможности создания нового метода защиты, сочетающего в себе преимущества приведенных методов, выделены основные требования к методу и пути построения такого метода. Полученные в настоящей работе результаты могут быть использованы при разработке методов защиты Грид от вредоносного кода и инструментальных средств на основе таких методов.

СПИСОК ЛИТЕРАТУРЫ:

1. http://www.cern.ch.
2. http://lhc.web.cern.ch.
3. http://www.eu-egee.org/.
4. http://www.egee-rdig.ru/.
5. http://gridclub.ru/projects/grid2003.
6. *Ian Foster, Carl Kesselman*. The Grid. Blueprint for a new computing infrastructure. Morgan Kaufman, 1998.
7. http://gridclub.ru.
8. http://www.worldcommunitygrid.org/stat/viewGlobal.do.
9. *Wenbo Mao, Fei Yan, Chunrun Chan*. Daonity—Grid Security with Behaviour Conformity from Trusted Computing — ACM. New York, NY, USA, 2006. P. 43—46.
10. *Matthew Smith, Michael Engel, Thomas Friese, Bernd Freisleben*. Security Issues in On-Demand Grid and Cluster Computing // Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06). May 16—006. P. 24.
11. *Хухлаев Е. В.* Безопасность Грид-диспетчера, реализованная средствами Грид-служб // Международная конференция «Распределенные вычисления и Грид-технологии в науке и образовании. 29 июня — 2 июля 2004 г. Дубна, 2004.
12. *Howard Chivers*. Grid Security: Problems and Potential Solutions. Department of Computer Science, University of York. Heslington; York, 2003.
13. *Ali Raza Butt, Sumalatha Adabala, Nirav H. Kapadia, Renato J. Figueiredo, Jose A. B. Fortes*. Grid-computing portals and security issues. J. Parallel Distrib. Comput. 63 (2003). 1006—1014.
14. *Xin Zhao, Kevin Borders, Atul Prakash*. SVGrid: A Secure Virtual Environment for Untrusted Grid Applications // Proceedings of the 3rd international workshop on Middleware for grid computing. Grenoble, France, 2005. P. 1—6.
15. http://www.globus.org/Security/GSI3/GT3-Security-HPDC.pdf.
16. *Peng Li*. Safe Systems Programming Languages. Department of Computer and Information Science. University of Pennsylvania, 2004.
17. *Xavier Leroy*. Java bytecode verification: an overview // Proceedings of the 13th International Conference on Computer Aided Verification. London, 2001. P. 265—285.
18. *Кларк Э. М., Грамберг О. Мл., Пелед Д.* Верификация моделей программ: Model Checking. М., 2002.
19. *Vijay Saraswat*. Java is not type-safe // http://www.cis.upenn.edu/~bcpierce/courses/629/papers/Saraswat-javabug.html.

Д. А. Щелкунов

Московский государственный технический университет им. Н. Э. Баумана

МЕТОДИКА ЗАПУТЫВАНИЯ КОДА ДЛЯ АВТОМАТИЧЕСКОЙ ЗАЩИТЫ ПРИЛОЖЕНИЙ ОТ НЕЛЕГАЛЬНОГО РАСПРОСТРАНЕНИЯ

Рассматриваются теоретический аппарат, описывающий запутывание кода и данных, проблемы, стоящие перед разработчиками систем защиты программ от нелегального распространения, и существующие методы их решения. Рассматривается доказательство теоремы Барака о невозможности создания идеального обфускатора, приведены замечания автора по этому поводу. Предлагается метод запутывания кода и данных, позволяющий создать защиту, сравнимую по стойкости с виртуальными машинами, но обеспечивающий гораздо более высокое быстродействие запутанного кода и данных.

Особое направление в развитии систем защиты составляют утилиты автоматической защиты приложений, т. е. такие программы, которые, не требуя от пользователя специфичных знаний, позволяют внедрить в тело защищаемой подпрограммы механизм, осуществляющий защиту приложения. До недавнего времени считалось, что такого рода «навесные защиты» не являются достаточно стойкими к взлому и зачастую «снимаются» гораздо проще, чем защиты, разработанные специально для конкретного приложения. В самом деле, большинство утилит автоматической защиты работают достаточно стандартно. Естественно, что для взлома таких защит были созданы