**Электронный научный журнал "Математическое моделирование, компьютерный и натурный эксперимент в естественных науках"**
**http://mathmod.esrae.ru/**
URL статьи: **mathmod.esrae.ru/56-236**
**Ссылка для цитирования этой статьи:**
Власенко Д.В. Графовые модели в задачах анализа и обеспечения информационной безопасности // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках. 2026. №2

УДК 004.056.5 DOI:10.24412/2541-9269-2026-2-24-31

# ГРАФОВЫЕ МОДЕЛИ В ЗАДАЧАХ АНАЛИЗА И ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Власенко Д.В.
Саратовский государственный технический университет имени Гагарина Ю.А.,
Россия, Саратов, d4nila.vlasenko@yandex.ru

# GRAPH MODELS IN PROBLEMS OF ANALYSIS AND ASSURANCE OF INFORMATION SECURITY

Vlasenko D.V.
Yuri Gagarin State Technical University of Saratov, Russia, Saratov,
d4nila.vlasenko@yandex.ru

**Аннотация.** В настоящее время графовые модели находят широкое применение в области моделирования информационных систем, систем защиты информации. В данной статье рассматривается применение теории графов в области информационной безопасности, их основные направления. Для каждого из них проводится обзор основной идеи метода, способ построения графа и решаемые им задачи.

**Ключевые слова:** теория графов, граф атак, информационная безопасность, защита информации, модель доступа.

**Abstract:** Currently, graph models are widely used in the field of information system modeling and information protection systems. This article explores the application of graph theory in the field of information security and its main areas. For each area, it provides an overview of the main idea of the method, the way of constructing the graph, and the tasks it solves.

**Keywords:** graph theory, graph of attacks, information security, information protection, access model.

**Введение**

На текущий момент количество угроз информационной безопасности не только не уменьшается, но и неуклонно растет. Это связано не только с цифровизацией многих сфер деятельности, но и развитием тактик нарушителей и усложнением сценариев атак, особенно в области социальной инженерии. Концепция эксплуатации одной конкретной уязвимости всё чаще стала заменяться многошаговыми сценариями. Это ситуации, при которых нарушитель находит сразу несколько слабых мест системы и перемещается между ними, превращая одиночную атаку в каскадный эффект. Пользователь обращается к файлу, открывается сетевой пор, злоумышленник получает доступ к серверу. Подобные сценарии описываются не изолированными объектами, а отношениями между ними.

Одним из современных подходов описания подобных комплексных сценариев являются графовые модели. Их основная идея заключается в том, чтобы представить сложную систему в виде графа, вершинами которого могут выступать элементы системы (например, файлы, компьютеры, процессы), а ребрами считаться связи и отношения между ними (например, права доступа, сетевые взаимодействия и т.д). Подобные модели позволяют оценивать весь путь нарушителя целиком, а не только фиксировать отдельные уязвимости.

Благодаря подобным возможностям, графовые модели находят применение во множестве областей информационной безопасности. Например, в формальном анализе прав доступа, построении графа атак до обнаружения аномалий в поведении системы или при расследовании инцидентов информационной безопасности.

Целью данной статьи является обзор таких направлений, и описание их основных идей.

**Первые простейшие модели**

Использование графов для описания информационных систем началось ещё в конце XX века, когда компьютерные сети только начали свое распространение. В то время появились первые модели, представлявшие доступ к ресурсам в виде направленного графа, но связано это было не с сетевыми базами данных, а с защитой информации в ОС.

С тех пор графовый подход прошёл путь от теоретических моделей до практических инструментов, которыми сегодня пользуются многие специалисты, аналитики, администраторы.

Первой моделью считается работа Батлера Лампсона, в которой он показал, что матрицу доступа можно представить в виде направленного графа [1]. Дуги в нем были помечены правами доступа. Например, дуга от субъекта к файлу с меткой «read» означает, что субъект может читать этот файл. Такой

подход позволил наглядно ответить на основные вопросы системы о том, какие файлы какому сотруднику доступны, не появились ли у кого-нибудь избыточные права.

Однако данная модель показывает только текущее состояние прав, в то время как в реальной системе права могут передаваться. Для учета подобной динамики была разработана модель Take-Grant [2]. Эта модель развила идею графов Лампсона до строгой математической системы. Система также описывается направленным графом, но в отличие от работы Лампсона, в данной модели используются 4 права доступа помимо read и write: право take (взять) и grant (предоставить). Если пользователь А имеет право take на пользователя Б, он может «забрать» себе любое право, которым обладает Б, а если А имеет право grant на Б, он может «передать» Б любое своё право на кого-то ещё. Граф при этом становится динамическим, поскольку он отражает передачу полномочий внутри систему.

Главным отличием этой модели от предыдущей стало то, что по исходному графу можно определить, возможна ли утечка информации. Например, может ли пользователь А получить доступ к объекту С. Если подобный путь обнаруживается, значит система содержит скрытую угрозу, даже если кажется, что все права настроены верно.

Модель Take-Grant в чистом виде используется крайне редко, но именно она указала на важность распространения прав в системе.

**Модели графов атак**

Помимо моделей по типу Take-Grant, описывающих в теории права доступа, существуют и графы, показывающие пути с точки зрения нарушителя. Они называются графами атак - графы, которые представляют все возможные последовательности действий злоумышленника (нарушителя) для достижения поставленных целей (угроз). Граф атак отвечает на вопрос о том, могут ли уязвимости системы эксплуатироваться совместно. Он создает картину цепочки шагов, по которым может следовать злоумышленник.

Считается, что первыми такими моделями были не вышеописанные графы, а деревья отказов (FTA). Бралось нежелательное событие, которое являлось корнем, и строилось дерево, показывающее, какие комбинации могут к нему привести. В 1999 году Брюс Шнайер опубликовал работу, посвященную дереву атак [3]. При этом подходе корень дерева в виде случайного отказа из модели дерева отказов заменялся на цель злоумышленника. Промежуточные узлы связывались логическими операторами «И» и «ИЛИ». Если узлы соединены оператором «И», то злоумышленнику необходимо выполнить все дочерние шаги, в случае соединения оператором «ИЛИ» - любой из них.

Главным недостатком деревьев было то, что они описывали лишь один вероятный сценарий, поэтому им на смену пришли графы. Они отлично моделировали сеть, в которой злоумышленник может двигаться различными

путями, а его цели могут пересекаться. Узлами такого графа стали не сценарии, а состояния системы, дугами являлись атаки, которые переводят систему из одного состояния в другое [4,5]. Такой подход позволил автоматизировать построение графа на основе данных об уязвимостях и топологии сети.

Данные работы стали основой для современных инструментов. Такие модели могут самостоятельно собирать информацию об инфраструктуре, строить граф и выявлять уязвимые маршруты [6]. Для построения такого графа сначала задается начальная точка входа, которой может воспользоваться злоумышленник. Затем строятся вершины графа, которые могут быть, например, узлы сети, права доступа и т.д. Ребрами же обозначаются возможные переходы между этими элементами. Как правило, в настоящее время такие графы строится автоматически.

Одним из самых известных современных инструментов является BloodHound, позволяющий анализировать и визуализировать связи в Active Directory. Он собирает данные о пользователях, группах, компьютерах, активных сессиях и правах доступа, после чего отображает их в виде графа. На таком графе можно наглядно увидеть путь от рядового пользователя до администратора домена, а также уязвимости в конфигурации, включая избыточные права доступа, небезопасные списки контроля доступа и проблемы с вложенностью групп. Всё это помогает специалистам по информационной безопасности обнаруживать сложные последовательности атак, которые сложно выявить стандартными методами.

Таким образом, графы атак решают сразу несколько задач информационной безопасности.

1) Поиск кратчайшего пути от точки входа злоумышленника до конечной цели. Математически каждое ребро имеет вес, отражающий сложность прохождения по нему или вероятность успешной эксплуатации уязвимости. Кратчайший путь укажет наиболее вероятный сценарий атаки.

2) Поиск минимального набора критических уязвимостей. В случае отсутствия возможности закрыть все уязвимости одновременно, требуется определить, какие из них следует устранить в первую очередь. Для этого необходимо найти такой набор ребер минимальной стоимости, удаление которого уничтожит маршруты от точки входа злоумышленника до конечной цели.

3) Оценка вероятности эксплуатации. В случае, если на ребрах графа задана вероятность успешной эксплуатации уязвимостей, возможно определить итоговую вероятность того, что злоумышленник дойдет до конечной цели хотя бы по одному из маршрутов. Такой подход позволяет ранжировать угрозы в порядке приоритета, что особенно полезно при ограниченных ресурсах информационной безопасности какой-либо организации.

## Модели графов для обнаружения атак и аномалий

Графы атак описывают потенциальные угрозы, а значит только то, что при определенных условиях может произойти. Однако графовые модели также могут выполнять и задачи выявления аномалий и отклонений от нормального поведения за счет анализа событий, которые уже происходят в системе.

До сложной математики использовалась простая форма графа, основанная на линейной концепции или цепочки. События сопоставлялись по времени и идентификаторам, а связи между ними восстанавливались вручную. Поэтому часто сами маршруты ломались, если злоумышленник менял порядок действий или делал паузы.

К концу 1990-х появилась идея графа взаимодействия. Такой граф уже не был линейным, а позволял отразить разветвленную структуру атаки благодаря тому, что данные начали анализировать не как поток событий, а как структуру взаимодействий. В таком случае один процесс может стать основой для появления нескольких дочерних. Появились исследования, в которых нестандартные случаи искали не по содержимому событий, а по форме того, как вершины и ребра графа складываются в характерные паттерны. Тогда подобные метрики стали применяться для выявления подозрительных узлов. Одной из первых разработок была модель Graph-based Intrusion Detection System (GrIDS) Калифорнийского университета в Дэвисе [7]. Данная модель представила сетевую активность в виде графа деятельности, узлами которого были хосты. Ребра же представляли собой сетевые соединения. Граф строился в реальном времени, аномалии находились не по содержанию пакета, а по форме самого графа. Например, модель находила атаку, обнаруживая аномальный рост числа соединений, что было невозможно заметить, глядя в плоский лог.

Следующий шаг развития подобных графом произошел во время роста числа современных целевых атак. Это сложные киберугрозы, направленные на конкретные организации, людей или инфраструктуру с целью кражи данных, шпионажа, нарушения деятельности или получения финансовой выгоды. Такие атаки могут длиться неделями, месяцами или даже годами. Злоумышленники тщательно изучают инфраструктуру жертвы, собирают информацию о средствах защиты, сотрудниках, используемом ПО. Для их обнаружения необходимо анализировать поведение системы на протяжении большого промежутка времени. Тогда появились графы зависимостей, в которых фиксируются причинно-следственные связи между событиями [8]. Если один процесс записал данные в файл, а другой его прочитало, то между ними образовалась информационная зависимость. Такие графы позволяют отследить маршрут данных от точки входа злоумышленника до момента достижения цели, даже в случае большой разницы во времени между этими событиями. Многие методы используют алгоритм поиска кратчайших путей, чтобы связать события через цепочку скомпрометированных процессов.

В настоящее время все три вышеописанных концепции объединяются в современных системах анализа и выстраивают полную картину происходящего

в информационной системе [9]. Результатом этого является формирование модели активности системы, на которой аномалии проявляются более отчетливо, чем в потоке отдельных записей.

Подобные графы решают несколько задач:

1) Поиск известных паттернов. Многие действия злоумышленника имеют характерные черты. Например, модель Kill Chain описывает атаку как последовательность этапов, начиная от разведки и заканчивая достижением цели [10]. На этапе управления зараженный источник обычно устанавливает периодические соединения с командным центром. Граф сетевых взаимодействий в таком случае выглядит как множество ребер, которые связывают узел с одним и тем же IP-адресом в разные моменты времени. Задача обнаружения такой активности сводится к поиску пути, соответствующего шаблону Kill Chain.

2) Вычисление метрик центральности. Такие метрики позволяют выявить аномалии без заданных шаблонов. Степень вершины при резком скачке может указывать на сканирование сети или активность вредоносного кода. Или, например, если пользовательский узел начинает демонстрировать аномально высокую активность, это может означать, что злоумышленник использует его для перемещения между сегментами сети.

3) Визуализация. Как правило, графы также выступают и средством визуального представления действий нарушителя. Например, платформы Maltego или MISP позволяют строить связи между индикаторами компрометации, IP-адресами, учетными записями и т.д., а графовые системы управления базами данных применяются для анализа таких связей в больших масштабах [11]. Визуализация графа позволяет построить полную картину инцидента, на восстановление которой по журналам ушли бы часы.

## Заключение

В рамках данной статьи были рассмотрены основные графовые модели, применяемые в сфере информационной безопасности. Показано, что основным преимуществом графовых моделей является то, что возможно анализировать не отдельные элементы системы, а связи между ними, что позволяет выявлять скрытые пути распространения атак, которые при классических методах анализа остаются незамеченными. Поэтому дальнейшее развитие инструментов на их основе в наше время всё так же остается актуальной задачей.

## Библиографический список

1. Lampson B.W. Protection // Proceedings of the 5th Princeton Symposium on Information Sciences and Systems. — Princeton, 1971. — P. 437–443. – URL: https://www.researchgate.net/publication/2414680_Protection (Дата обращения 29.05.2026)
2. Богульская, Н. А. Модели безопасности компьютерных систем : учебное пособие / Н. А. Богульская, М. М. Кучеров. — Красноярск : СФУ, 2019. —

206 с. — ISBN 978-5-7638-4008-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/157578 (дата обращения: 29.05.2026).

3. Schneier B. Attack Trees: Modeling Security Threats // Dr. Dobb's Journal. — 1999. — Vol. 24, No. 12. — P. 21–29. — URL: https://www.schneier.com/academic/archives/1999/12/attack_trees.html (дата обращения 30.05.2026).
4. Sheyner O., Haines J., Jha S., Lippmann R., Wing J.M. Automated Generation and Analysis of Attack Graphs // Proceedings of the IEEE Symposium on Security and Privacy. — Oakland, 2002. — P. 273–284. URL: https://ieeexplore.ieee.org/document/1004377 (дата обращения: 30.05.2026)
5. Ammann P., Wijesekera D., Kaushik S. Scalable, Graph-Based Network Vulnerability Analysis // Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS). — Washington, 2002. — P. 217–224. URL: https://www.researchgate.net/publication/221609714_Scalable_graph-based_network_vulnerability_analysis (дата обращения 30.05.2026)
6. Noel S., Jajodia S. Managing Attack Graph Complexity Through Visual Hierarchical Aggregation // Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security (VizSec). — Washington, 2004. — P. 109–118. URL: https://www.researchgate.net/publication/221325889_Managing_attack_graph_complexity_through_visual_hierarchical_aggregation (дата обращения 30.05.2026)
7. Staniford-Chen S., Cheung S., Crawford R., Dilger M., Frank J., Hoagland J., Levitt K., Wee C., Yip R., Zerkle D. GrIDS — A Graph-Based Intrusion Detection System for Large Networks // Proceedings of the 19th National Information Systems Security Conference. — Baltimore, 1996. — P. 361–370. —URL: https://www.academia.edu/85404023/GrIDS_A_graph_based_intrusion_detection_system_for_large_networks (дата обращения 30.06.2026)
8. King S.T., Chen P.M. Backtracking Intrusions // Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP). — Bolton Landing, 2003. — P. 223–236. URL: https://pdos.csail.mit.edu/archive/6.824-2004/papers/king03.pdf (дата обращения 30.06.2026)
9. Hossain M.N., Milajerdi S.M., Wang J., Eshete B., Gjomemo R., Sekar R., Stoller S., Venkatakrishnan V.N. SLEUTH: Real-time Attack Scenario Reconstruction from COTS Audit Data // Proceedings of the 26th USENIX Security Symposium. — Vancouver, 2017. — P. 487–504. URL: https://www.researchgate.net/publication/318249689_SLEUTH_Real-time_Attack_Scenario_Reconstruction_from_COTS_Audit_Data (дата обращения 30.06.2026)

10. Hutchins E.M., Cloppert M.J., Amin R.M. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains // Proceedings of the 6th International Conference on Information Warfare and Security. — Washington, 2011. — P. 113–125. URL: https://www.ciosummits.com/media/solution_spotlight/LM_Cyber_Kill_Chain_White_paper_2011.pdf (дата обращения 30.06.2026)
11. Wang W., Daniels T.E. A Graph Similarity-Based Approach to Security Event Analysis Using Correlation Techniques // Proceedings of the 13th IEEE International Workshop on Information Assurance (IWIA). — 2011. — URL: https://www.researchgate.net/publication/221285596 (дата обращения: 31.05.2026).