БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Искусственные иммунные системы и их применение / Под ред. Д. Дасгупты; Пер. с англ. под ред. А.А. Романюхи. – М.: Физматлит, 2006. – 344 с.
2. *Kuby J.* Immunology. W.H. Freeman and Co., 2nd edition, 1994.
3. *Forrest S., Perelson A.S., Allen L., Cherukuri R.* Self-nonself discrimination in a computer // In: Proc. of IEEE symposium on research in security, Oakland, CA, 16-18 May, 1994. – P. 202-212.
4. *Dasgupta D., Forrest S.* Tool breakage detection in milling operations using a negative-selection algorithm // Technical report CS95-5, Department of computer science, University of New Mexico, 1995.
5. *Percus J.K., Percus O., Perelson A.S.* Predicting the size of the antibody combining region from consideration of efficient self/non-self discrimination // PNAS. – 1993. – Vol. 60. – P. 1691-1695.
6. *D'haeseleer P., Forrest S., Helman P.* An immunological approach to change detection: algorithms, analysis, and implications // In: Proc. of Ieee symposium on research in security, Oakland, CA, May, 1996.
7. *Forrest S., Hofmeyr S.A. Somayaji A., Longstaff T.A.* A sense of self for unix processes // In: Proc. Of IEEE symposium on research in security and privacy, Oakland, CA, May, 1996.
8. *Bersini H., Varela F.* The immune learning mechanisms: Recruitment reinforcement and their applications // Computing with biological metaphors (Ed/ R/ Patton). – L.: Chapman and Hall, 1994.
9. *Jerne N.K.* Towards a network theory of the immune system // Ann. Immunol. (Inst/ Pasteur). – 1974. – Vol. 125. – P. 435-441.

Статью рекомендовал к опубликованию к.т.н. М.Ю. Руденко.

Брюхомицкий Юрий Анатольевич – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: bya@tsure.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634371905; кафедра безопасности информационных технологий; доцент.

Bryukhomitsky Yuriy Anatoly – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education "Southern Federal University"; e-mail: bya@tsure.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634371905; the department of security in data processing technologies; associate professor.

УДК 004.056; 004.8

Е.С. Абрамов, А.В. Андреев, Д.В. Мордвин

ПРИМЕНЕНИЕ ГРАФОВ АТАК ДЛЯ МОДЕЛИРОВАНИЯ ВРЕДОНОСНЫХ СЕТЕВЫХ ВОЗДЕЙСТВИЙ

Использование графов атак при проведении анализа защищённости позволяет учесть взаимосвязи отдельных узлов и их параметры защищенности, что даёт более точные данные для оценки защищенности всей системы в целом, чем при исследовании свойств защищенности отдельных узлов.

Эта статья описывает процесс расчета графа атак, анализ полученных результатов и оценку эффективности существующих контрмер.

Модель сети уточнена до уровня сервисов, а не пары <интерфейс-порт>. В модели учтены динамическая маршрутизация и фильтрация на любом сетевом объекте, NAT, состояния в графе атак детализированы до триады <конфиденциальность, целостность, доступность>. При построении графа атак учитываются и локальные, и сетевые уязвимости.

Представлены результаты экспериментальной оценки производительности системы. Для анализа 10 000 моделируемых хостов потребовалось в среднем около 100 секунд.

Количество правил политики разграничения доступа (от 500 до 4000 в каждой моделируемой подсети) подбиралось таким образом, чтобы максимальное количество правил фильтрации на устройствах составляло около 1000.

Графы атак; анализ защищённости; эффективность контрмер; имитационное моделирование; NetSPA; ISO/IEC 15408.

E.S. Abramov, A.V. Andreev, D.V. Mordvin

EVALUATION OF CORPORATE NETWORKS SECURITY BASED ON ATTACK GRAPHS

Using attack graphs for the security analysis allows to consider the relationship of individual components and their security parameters. It gives more accurate data to assess the security of the system as a whole comparing with investigation of security properties of the individual nodes. This paper describes the calculation of attack graph, analyze the results and evaluate the effectiveness of existing countermeasures. The model allows dynamic routing, filtering on any network object, NAT. States in attack graph are detailed to <confidentiality, integrity, availability> triad. In constructing the attack graph takes into account both local and network vulnerability. The results of experimental evaluation of system performance presented. For the analysis of 10000 simulated hosts took an average time of about 100 seconds. The number of access control rules (from 500 to 4000 per simulated subnet) were chosen so that the maximum number of filtering rules for devices were about 1,000.

Attack graph; security analysis; countermeasures effectiveness; computer simulation; NetSPA; ISO/IEC 15408.

Введение. Основой оценки безопасности ИТ-систем и продуктов является стандарт ISO/IEC 15408. Основой философии ISO/IEC 15408 является активное исследование всей ИТ-системы, которой необходимо доверять [1]. Центральным компонентом процесса исследования безопасности систем на данный момент являются сканеры уязвимостей. Недостатком использования сканеров уязвимостей является то, что исследование свойств защищенности отдельных узлов системы еще не дает достаточно информации для оценки защищенности всей системы в целом. Учет взаимосвязи отдельных узлов и их параметров защищенности может быть получен за счет использования графов атак. Несмотря на то, что работы в данной области ведутся уже достаточно давно, графы атак пока являются в большей степени исследовательскими проектами.

В работе рассматриваются все фазы функционирования программного обеспечения, предназначенного для проведения оценки защищённости: автоматизация построения модели сети, расчета графа атак, анализа полученных результатов и эффективности существующих контрмер, автоматизации процесса совершенствования и выработки новых контрмер.

В нашей работе, по отношению к уже существующим наиболее актуальным работам [2, 3], можно выделить следующие особенности:

- модель уточнена до уровня сервисов, а не пары интерфейс-порт;
- в модели учтены динамическая маршрутизация и фильтрация на любом сетевом объекте (см. п. 2);
- состояния в графе атак детализированы до триады конфиденциальность, целостность, доступность (подробнее граф атак будет рассмотрен в п. 3);
- при построении графа атак учитываются и локальные, и сетевые уязвимости.

Далее в статье будут рассмотрены модель сети и данные, на основе которых она строится, процесс построения графа атак и визуализации результатов. В конце статьи будут рассмотрены будущие направления работ.

1. Модель сети. При разработке модели мы выделили следующие основные черты реальных сетей, которые необходимо моделировать:

- наличие статической и динамической маршрутизации;
- возможность фильтрации на любом сетевом узле;
- поддержка преобразований адресов источника и получателя (SNAT и DNAT);
- уявзимости соответствуют сервисам (под сервисом мы понимаем любой программный компонент);
- сервисы могут быть локальными и сетевыми.

Разработанную нами статическую модель сети иерархически можно представить следующим образом:

- подсеть: соответствует подсети в реальной сети, включает в себя сетевые объекты;
- сетевой объект: рабочая станция, коммутатор, маршрутизатор или сервер. Для любого сетевого объекта может быть определена маршрутизация, фильтрация, заданы сетевые интерфейсы, определены сервисы;
- маршрутизация: поддерживается статическая и динамическая маршрутизация. Формат правил аналогичен формату правил в linux;
- фильтрация: поддерживается пакетная фильтрация, SNAT и DNAT. Возможны два варианта фильтрации по умолчанию: разрешить, запретить. Формат правил аналогичен формату правил iptables;
- сетевой интерфейс: для интерфейса определяется ip-адрес, маска подсети, шлюз по умолчанию;
- сервис: определяется на основе CPE-идентификатора [4]. При этом из базы уязвимостей и сервисов автоматически выбираются уязвимости, соответствующие данному сервису. Любая из уязвимостей может быть помечена как исправленная. Сервис может быть локальным или сетевым. Для сетевого сервиса должна быть определена одна или более оконечных точек. Оконечная точка представляет собой пару ip-порт. Существуют возможность добавить свой сервис (без CPE-идентификатора). Для любого сервиса вручную может быть добавлена своя уязвимость с параметрами, соответствующими базовым метрикам CVSS [5]. Таким образом, может быть смоделирован любой сервис с произвольным набором различных уязвимостей, в том числе уязвимостей нулевого дня.

Модель может строиться как в полуавтоматическом режиме на основе отчетов сканеров уязвимостей (Nessus, OpenVAS и т.п.), OVAL-сканера, так и вручную в разработанной среде. Основная проблема использования сканеров уязвимостей заключается в том, что они направлены на обнаружение именно уязвимостей, а не сервисов. На обнаружение сервисов и их версий направлен NMAP, но структура его правил такова, что связать обнаруженные сервисы и версии с CPE-идентификаторами практически невозможно. Поэтому мы сейчас ведем работу над автоматизированной подсистемой обработки информации из разных источников и представлением результатов в виде CPE-идентификаторов.

Одну из ключевых ролей при построении и эксплуатации модели играет база сервисов и уязвимостей. При создании базы нами были использованы National Vulnerability Database (NVD) [6] и Official Common Platform Enumeration (CPE) Dictionary [7]. NVD содержит достаточно полную информацию по уязвимостям и конфигурации окружения, в которых эти уязвимости могут быть проэксплуатированы. Для исследовательских целей на текущем этапе этих данных нам было достаточно. Для создания законченного продукта в данный момент мы ведем работы по совершенствованию словаря CPE, а также рассматриваем возможности сбора дополнительных данных, например, по номерам портов, используемых сервисами по умолчанию, и данных о зависимостях между сервисами.

2. Граф атак и анализ результатов. Как и в работе [2], перед построением графа атак мы вычисляем матрицу достижимости. Но в нашей работе достижимость рассчитывается не между объектами и парой ip-порт, а между объектами сети и сервисами. При этом при расчете матрицы достижимости учитываются следующие особенности:

- динамическая маршрутизация учитывается следующим образом: сервис считается достижимым, если он достижим хотя бы по одному маршруту;
- возможность наличия нескольких оконечных точек для сервиса учитывается следующим образом: сервис считается достижимым, если он достижим хотя бы на одной оконечной точке;
- при расчете матрицы учитываются возможности преобразования адресов источника и получателя.

Таким образом, мы всегда учитываем наихудший вариант (сервис достижим) в контексте решаемой задачи оценки защищенности.

Далее строится граф атак, узлами которого являются состояния сетевых объектов, ребрами – уязвимости, вследствие эксплуатации которых совершился переход из состояния 1 в состояние 2 (рис. 1). Состояния в графе кодируются в виде триплета влияния на доступность, конфиденциальность, целостность. Значения влияний базируются на стандарте CVSS [5] и могут быть следующими: отсутствует, частичное, полное. Таким образом, максимальное количество состояний для одного сетевого объекта – 27. При этом алгоритм расчета графа атак построен таким образом, что набор состояний может быть достаточно просто изменен.

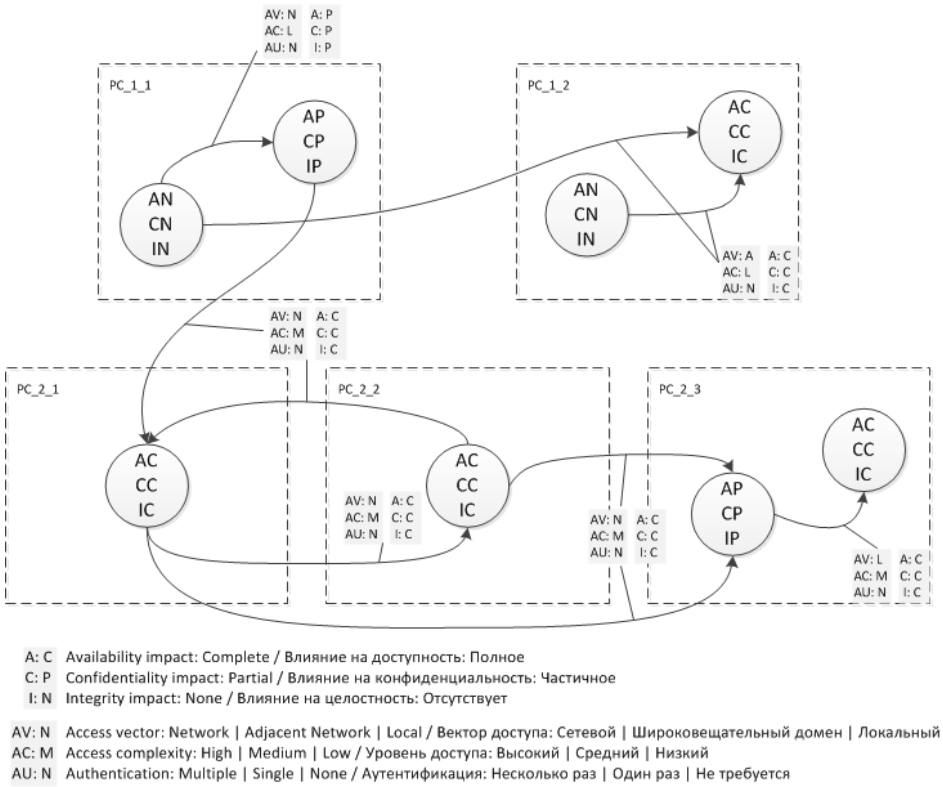

Рис. 1. Пример графа атак

Граф атак может строиться в одном из следующих режимов:

- из одного объекта-источника атаки;
- конечного набора объектов-источников атаки;
- всех возможных объектов-источников атаки.

Помимо расчета самого графа отдельно строится список маршрутизаторов и межсетевых экранов, для которых есть хотя бы одно состояние полного влияния на целостность. В результате проведения атак на эти сетевые объекты злоумышленник может существенно изменить достижимость в сети.

Сам по себе граф атак является компактным хранилищем всех возможных атак в данной модели сети. В таком виде он нигде не визуализируется. На данном этапе нами предусмотрены два режима отображения результатов:

- в виде списка всех возможных атак;
- в виде таблицы статистики по состояниям.

При отображении списка всех возможных атак каждая отдельная атака может быть отображена на модели. Кроме того, для нее отображается подробная информация по каждому этапу атаки (рис. 2).

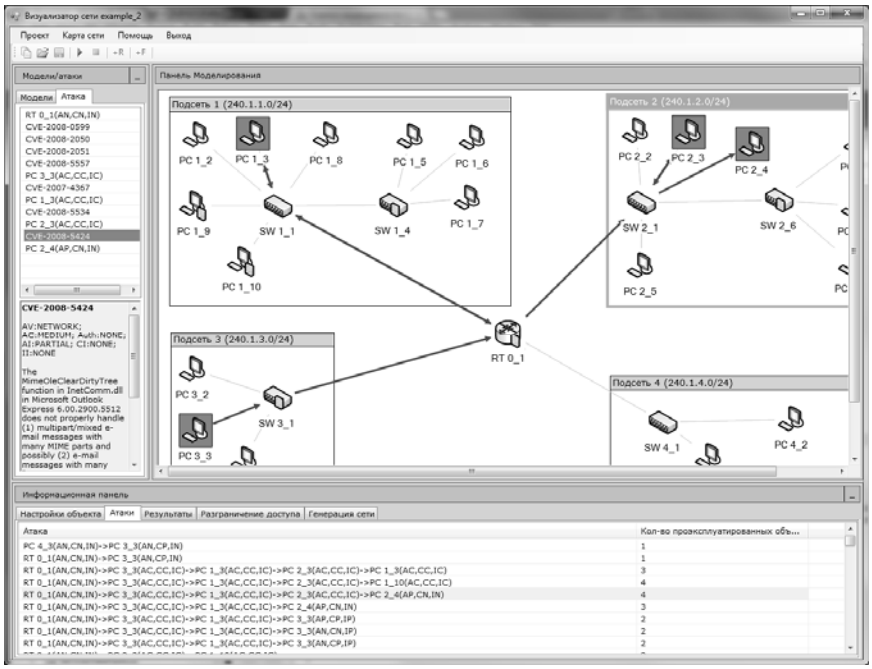

Рис. 2. Визуализация этапов атаки

Список всех возможных атак получается очень большим, поэтому сейчас мы рассматриваем необходимость введения в модель активов (в терминологии ISO/IEC 15408 [8]) и предоставления на их основе возможности сортировки и фильтрации списка атак. Помимо этого фильтровать список возможных атак можно на основе объекта, выбранного в таблице статистики по состояниям (рис. 3).

При отображении статистики по состояниям для каждого проэксплуатированного объекта отображается общее количество и количество проэксплуатированных уязвимостей. Кроме того, подробно по каждому состоянию показывается, какое количество уязвимостей приводит к возникновению у объекта конкретного состояния.

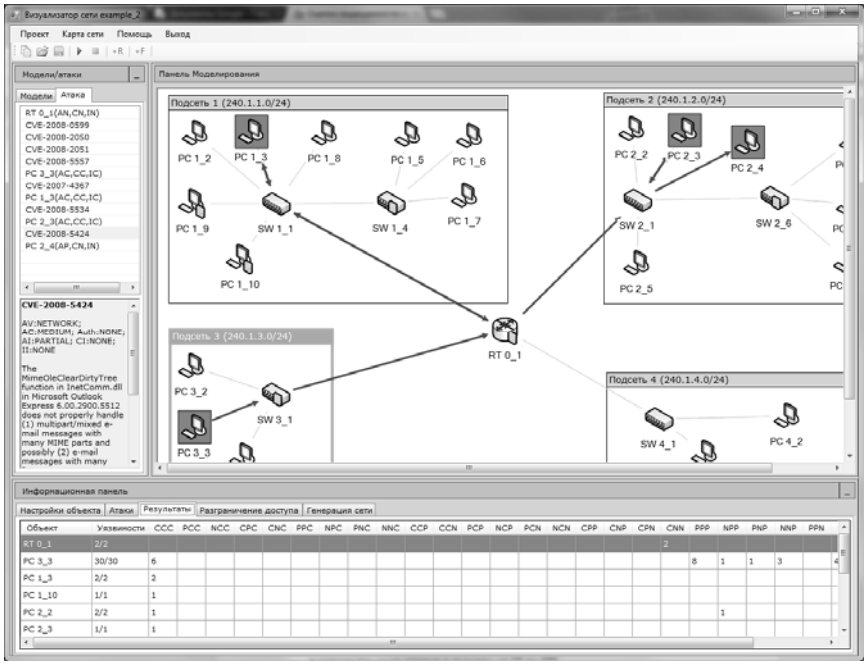

Рис. 3. Таблица статистики по состояниям

Помимо этого существует возможность отображения достижимости от сетевого объекта и к сетевому объекту (рис. 4).

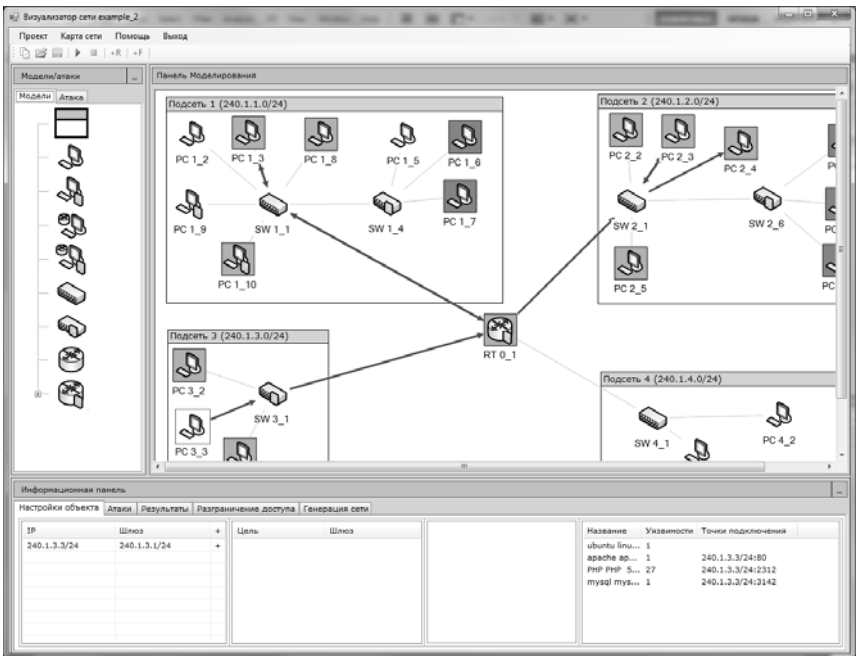

Рис. 4. Достижимость от сетевого объекта PC 3_3

3. Экспериментальная оценка производительности. Для оценки производительности разработанных алгоритмов проводились расчеты на сгенерированных сетях различной размерности (рис. 5). При генерации сетей варьировались следующие параметры:

- количество подсетей: от 15 до 200;
- количество компьютеров в подсети: от 10 до 100;
- количество коммутаторов в подсети: от 1 до 3;
- количество роутеров с фильтрацией: от 5 до 10;
- на каждом компьютере добавлялся один сетевой и один локальный сервис с двумя уязвимостями каждый, параметры уязвимости генерировались случайным образом на базе CVSS-метрик.

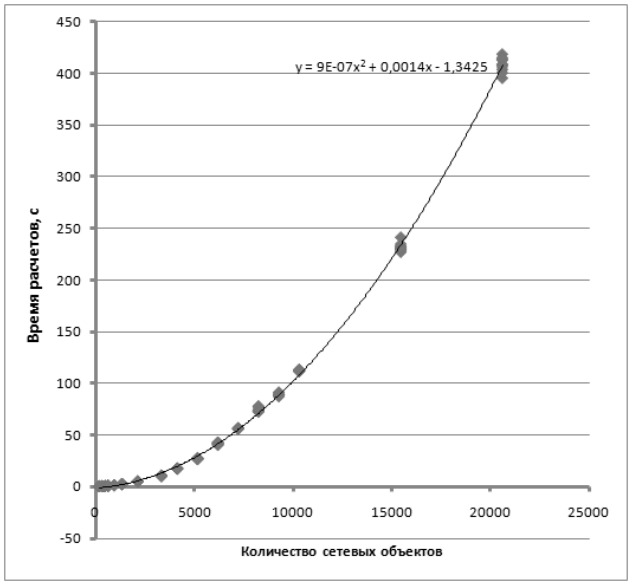

Рис. 5. График зависимости совокупного времени расчетов графа атак и матрицы достижимости от количества сетевых объектов

В процессе добавления маршрутизаторы соединялись между собой последовательно в одну линию. Затем каждый маршрутизатор подключался к случайной подсети. Таким образом, пути между двумя подсетями могут содержать как 1, так и все маршрутизаторы сети.

Для каждой сети в зависимости от ее размера генерировались от 500 до 4000 правил политики разграничения доступа. При этом соотношение правил устанавливалось следующим образом: 30 % правил – многие ко многим, 30 % правил – один ко многим и 40 % правил – один к одному. Правила межсетевых экранов генерировались на основе правил политики разграничения доступа с использованием алгоритмов, разработанных в работе [9]. Благодаря этому сгенерированные правила получаются гарантировано безошибочными. Количество правил политики разграничения доступа подбиралось таким образом, чтобы максимальное количество правил фильтрации на устройствах получалось около 1000.

Сети генерировались 10 раз для каждого набора параметров. Граф рассчитывался в режиме “из всех возможных объектов-источников атаки”. В результате для каждого эксперимента вычислялись следующие метрики:

- время расчета матрицы достижимости в миллисекундах;

- коэффициент достижимости в сети: вычислялся как отношение количества ячеек с положительной достижимостью к общему количеству ячеек матрицы;
- количество правил фильтрации на каждом устройстве с межсетевым экраном;
- время расчета графа атак в миллисекундах;
- количество состояний в графе атак;
- количество ребер (проэксплуатированных уязвимостей) в графе атак;
- количество непроэксплуатированных возможностей.

Совокупность данных метрик дает достаточно полную количественную картину каждого эксперимента (табл. 1).

Таблица 1

Экспериментальные данные для 10 310 сетевых объектов

Время расчета графа атак, мс	Количество состояний	Количество проэксплуатированных уязвимостей	Количество непроэксплуатированных уязвимостей	Время расчета матрицы достижимости, мс	Коэффициент достижимости	Количество правил на межсетевых экранах
9657	21458	1716810	1108281	101548	0,07362	181, 560, 698, 855, 909, 1005, 1005, 1035, 1066, 1188
9704	22493	2065050	1739193	101723	0,07014	296, 581, 702, 773, 927, 981, 1000, 1031, 1060, 1062
9751	23050	2200671	2205374	102933	0,068964	181, 454, 599, 788, 960, 973, 981, 1013, 1056, 1100
9580	23354	2111495	2495162	102226	0,060495	277, 377, 701, 730, 736, 906, 948, 976, 1010, 1051
9411	23552	1973906	2666644	103738	0,053225	253, 417, 579, 818, 846, 853, 923, 924, 1019, 1063
9628	23539	1917530	2844492	102442	0,053111	226, 463, 567, 698, 808, 894, 936, 959, 964, 985
9473	23630	1827738	2939579	102201	0,047632	214, 443, 471, 767, 778, 808, 871, 900, 941, 1009
9562	23511	1794648	3000288	101898	0,045306	273, 393, 553, 670, 744, 777, 848, 854, 945, 946
9896	23619	1855583	2995687	102856	0,044026	229, 279, 447, 699, 742, 795, 821, 846, 974, 1047
9290	23702	1675710	2937674	102604	0,038589	233, 476, 498, 666, 686, 772, 827, 834, 921, 950

4. Будущие направления работы. На данный момент наиболее проработаны этапы работы с моделью сети и построения графа атак. Проведен ряд работ по оптимизации алгоритмов, при этом некоторые направления оптимизации оставлены на будущее. Мы считаем полученные на данный момент результаты удовлетворительными. Сейчас основная работа идет над этапом автоматизации построения модели и совершенствования базы сервисов и уязвимостей.

Отдельного рассмотрения заслуживает вопрос анализа существующих в сети контрмер и автоматизации предложений по их совершенствованию. Основная проблема моделирования систем обнаружения и предотвращения вторжений и систем антивирусной защиты заключается в очень слабой формализации функций этих систем и отсутствии критериев оценки качества их работы. Для многих систем антивирусной защиты существуют так называемые антивирусные энциклопедии, в которых подробно описаны механизмы функционирования конкретных вирусов и борьбы с ними. Но эти базы никак не стандартизированы и не обеспечена

связность с какими-либо другими базами и стандартами, например, CVE [10]. Базы большинства систем обнаружения и предотвращения вторжений никак не представлены в открытом доступе. Таким образом, системы антивирусной защиты и большинство систем обнаружения и предотвращения вторжений не могут быть количественно оценены потребителем и смоделированы в рамках научно-практических исследований.

На данный момент нами прорабатываются реализации фильтрации с учетом контекста (stateful packet inspection) и моделирования системы обнаружения вторжений Snort. Помимо этого нами реализован метод автоматической генерации правил пакетного фильтра, адекватных требуемой политике разграничения доступа в сети, и метод оптимального размещения межсетевых экранов в сети, основанный на использовании генетических алгоритмов [9]. Учитывая эти наработки, мы планируем реализовать метод автоматизации предложений по оптимизации размещения систем обнаружения вторжений в сети, алгоритмы анализа правил межсетевых экранов [11] и метод автоматизации предложений по их модификации с целью исключения найденных ошибок.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ISO/IEC 15408-3:2009.
2. *Kyle Ingols, Matthew Chu, Richard Lippmann, Seth Webster, Stephen Boyer.* Modeling Modern Network Attacks and Countermeasures Using Attack Graphs. Annual Computer Security Applications Conference, 2009. – P. 117-126.
3. *Sushil Jajodia, Steven Noel.* Topological Vulnerability Analysis // Advances in Information Security. – 2010. – Vol. 46, № 4. – P. 139-154.
4. Common platform enumeration. MITRE. http://cpe.mitre.org.
5. Common Vulnerability Scoring System. Forum of Incident Response and Security Teams, Common Vulnerability Scoring System-Special Interest Group. http://www.first.org/cvss/.
6. National Vulnerability Database. http://nvd.nist.gov/download.cfm.
7. Official Common Platform Enumeration Dictionary. http://nvd.nist.gov/cpe.cfm.
8. ISO/IEC 15408-1:2009.
9. *Abramov E., Mordvin D., Makarevich O.* Automated method for constructing of network traffic filtering rules. In Proceedings of the 3rd international conference on Security of information and networks (SIN '10). ACM, New York, NY, USA, 2010. – P. 203-211. DOI=10.1145/1854099.1854141 http://doi.acm.org/10.1145/1854099.1854141.
10. Common Vulnerabilities and Exposures. MITRE. http://cve.mitre.org/.
11. *L. Yuan et al.* “FIREMAN: A toolkit for FIREwall modeling and ANalysis,” in IEEE Symposium on Security and Privacy // IEEE Computer Society. – 2006. – P. 199-213.

Статью рекомендовал к опубликованию к.т.н., доцент Д.В. Мордвин.

Абрамов Евгений Сергеевич – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: abramoves@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371905; кафедра безопасности информационных технологий; к.т.н.; доцент.

Андреев Артём Викторович – e-mail: andreev.artem@gmail.com; кафедра безопасности информационных технологий; ведущий программист.

Мордвин Денис Валериевич – e-mail: neverminden@gmail.com; кафедра безопасности информационных технологий; к.т.н.; доцент.

Abramov Evgeny Sergeevich – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: abramoves@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371905; the department of security in data processing technologies; cand. of eng. sc.; associate professor.

Andreev Artem Viktorovich – e-mail: andreev.artem@gmail.com; the department of security in data processing technologies; senior programmer.

Mordvin Denis Valerievich – e-mail: neverminden@gmail.com; the department of security in data processing technologies; cand. of eng. sc.; associate professor.

УДК 681.03.245

Л.К. Бабенко, А.С. Кириллов

РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ АТАКИ НА ГОСТ Р34.11-94 С ИСПОЛЬЗОВАНИЕМ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ

Описываются особенности реализации алгоритмов атаки на функцию хеширования ГОСТ а также результаты исследований реализованных алгоритмов на предмет возможности полного решения задачи атаки на ГОСТ Р34.11-94. В результате исследований, было выяснено, что существующие алгоритмы не позволяют провести полную атаку на ГОСТ за приемлемое время, о чем свидетельствуют оценки, полученные на основании проведенных экспериментов, представленные в данной работе. Реализованные алгоритмы могут применятся с большим успехом для атаки на другие хеш-функции.

ГОСТ; функция хеширования; параллельные вычисления; атака прообраза; мультиколлизии.

L.K. Babenko, A.S. Kirillov

DEVELOPMENT AND ANALYSIS OF ALGORITHMS OF ATTACK ON THE GOST R34.11-94 USING MULTIPROCESSOR SYSTEM

This article describes features of implementation of algorithms of attack on the GOST hash function, and results of it's analysis concerning the possibility of solution task of GOST R34.11-94 attack. As a result of investigations it was found that the existing algorithms do not allow a full attack on the GOST in acceptable period of time, as indicated by estimates obtained on the basis of the experiments presented in this paper. Implemented algorithms can be applied with great success for attack on other hash functions.

GOST; hash function; parallel computing; preimage attack; multicollision.

Функции хеширования являются одним из основных методов криптографической защиты информации и используются в алгоритмах цифровой подписи, в прикладных системах с открытым ключом [1]. В данной работе задачей ставится реализация алгоритмов атаки на ГОСТ Р34.11-94, исследования их характеристик, возможности их параллельного выполнения, а также определение возможностей современных вычислительных ресурсов для реализации атаки прообраза на ГОСТ.

На сегодняшний день существует 2 основных типа атак на криптографические функции хеширования[2]:

1. Атака нахождения коллизии. Данная атака представляет собой нахождение такой пары M_1 и M_2, хеш-значение которых одинаково и сложность выполнения атаки меньше чем $2^{n/2}$, где n – длина хеш-значения в битах [2].
2. Атака нахождения прообраза [3] криптографической хеш-функции. Данная атака состоит в нахождении сообщения с заданным значением хеша. Существует два типа подобных атак [3]:
   - атака нахождения первого прообраза: по заданному значению хеш h найти такое сообщение M, что $H(M)=h$, где H – функция хеширования, сложность атаки не должна превышать 2^n;