9. *Mashkina I.V.* Upravlenie zashchitoy informatsii v segmente korporativnoy informatsionnoy sistemy na osnove intellektual'nykh tekhnologiy [Management of information security in the corporate information system based on intelligent technologies: Dr. of eng. sc. diss]. Ufa, 2009, 354 p.
10. *Torokin A.A.* Inzhenerno-tekhnicheskaya zashchita informatsii [Engineering and technical protection of information]: Uchebnoe posobie [textbook]. Moscow: Gelios ARV, 2005, 960 p.

Статью рекомендовал к опубликованию д.т.н., профессор О.Б. Макаревич.

Тулиганова Лилия Равилевна – Уфимский государственный авиационный технический университет; e-mail: tulegan@rambler.ru; 450000, Республика Башкортостан, г. Уфа, ул. К. Маркса, 12; тел.: +79374747997; кафедра вычислительной техники и защиты информации.

Павлова Ирина Александровна – e-mail: i_pavlova@list.ru; тел.: +79191489821; кафедра вычислительной техники и защиты информации.

Машкина Ирина Владимировна – e-mail: mashkina_vtzi@mail.ru; тел.: +79279277089; кафедра вычислительной техники и защиты информации.

Tuliganova Liliia Ravilevna – The Ufa State Aviation Technical University; e-mail: tulegan@rambler.ru; 12, K. Marx's street, Ufa, 450000, Russia; phone: +79374747997; the chair of computer facilities and information protection.

Pavlova Irina Aleksandrovna – e-mail: i_pavlova@list.ru; phone: +79191489821; the chair of computer facilities and information protection.

Mashkina Irina Vladimirovna – e-mail: mashkina_vtzi@mail.ru; phone: +79279277089; the chair of computer facilities and information protection.

УДК 004.056.57

Д.А. Катаргин

ОБНАРУЖЕНИЕ МУТАЦИИ УЯЗВИМОСТЕЙ В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

Рассматривается методика обнаружения уязвимостей в программном обеспечении (ПО). Актуальность связана с необходимостью своевременного обнаружения уязвимостей в программных продуктах для предотвращения утечки или порчи пользовательских данных. В качестве объекта исследования выступает продукт «MS Office», так как уязвимости, присутствующие в нем, непосредственно влияют на личные документы пользователя. Для обнаружения мутаций уязвимости выявлены модули ПО, которые подвергаются наиболее частой эксплуатации. Для этого используется база уязвимостей NVD и классификаторы уязвимостей CWE. Но ссылки на классификаторы не всегда присутствуют в базе уязвимостей, поэтому была проведена оценка достаточности покрытия для дальнейшего исследования, которая показала, что классификаторы CWE присутствуют в базе для более чем 80 % уязвимостей, начиная с 2009 г. На основании полученных классификаторов из базы NVD была построена карта уязвимостей, из которой были получены векторы атаки: «вверх» – эксплуатация старой уязвимости в новой версии ПО; «вдоль» – эксплуатация уязвимости в смежном модуле, которая распространяется на смежные продукты; «вниз» – эксплуатация уязвимости из новой версии ПО на старой. Также к данным векторам приведены экспериментальные результаты, которые подтвердили положение методики к обнаружению уязвимостей в ПО с закрытым исходным кодом.

Уязвимость; мутация; программное обеспечение; карта уязвимостей.

D.A. Katargin

DETECTION OF VULNERABILITIES MUTATION IN SOFTWARE

This article presents a method for vulnerabilities detection in software. The urgency associated with the need for timely detection of vulnerabilities in software products to prevent leakage or damage user data. The object of research is the «MS Office», because vulnerabilities that are present in it, have a direct impact on the user's personal documents. For detection of mutations we identified vulnerable software modules by using a database of vulnerabilities NVD and classifiers vulnerabilities CWE. But, because it is a reference to the classifiers are not always present in the database vulnerabilities, therefore we check of coverage which showed that the classifiers CWE and it present in the database for more than 80 % of the vulnerabilities since 2009. Based on the classification of the base NVD we construct vulnerability map and vectors of attack: "Up" – exploitation old vulnerability for new software version; "Along" – exploitation for adjoining module which extends to related products; "Down" – exploitation vulnerability for new version of the software on the old one. This attack vectors of the experimental results, which showed the consistency of the method to detect vulnerabilities in software with closed source.

Vulnerability; software; vulnerability map; mutation.

Введение. По информации из баз уязвимостей и баз эксплоитов, следует, что уязвимостям подвержен довольно широкий спектр программного обеспечения. Одними из особо критичных для конечного пользователя являются уязвимости в офисных программных продуктах, которые способствуют краже или повреждению конфиденциальной информации (документов).[1] Часть таких продуктов поставляются с закрытым исходным кодом, ввиду чего невозможно применить средства статического анализа, положительным качеством которых является высокая скорость обработки больших объемов данных.[2]

Таким образом, для ПО с закрытым исходным кодом может быть применим только динамический анализ, т.е. контроль выполнения программы на виртуальном и реальном процессоре. Однако существенным недостатком данного способа поиска уязвимостей является необходимость в генерации большого объема входных данных для обеспечения покрытия всего кода.[3] Так, если ПО обладает сложной архитектурой и имеет большое количество модулей, данный анализ может производиться довольно длительный промежуток времени. Кроме того, данный вид анализа требует задействовать значительные вычислительные ресурсы.

Проанализировав существующие подходы и средства [4] решения поставленной задачи, была разработана методика, которая основывается на выявлении уязвимостей, сохранившихся для какой-либо версии ПО или же мутировали так, что для их эксплуатации необходимо внести незначительные коррективы в существующие эксплоиты. Для этого используется собственная карта уязвимостей, полученная из базы данных NVD и классификаторов CWE.

Методика. Исходя из поставленных целей, была разработана методика, которая состоит из следующих этапов:

- построение карты уязвимостей;
- выделение часто встречающихся уязвимостей;
- проверка возможности эксплуатаций уязвимостей в близких версиях программного обеспечения эксплоитом с незначительными изменениями или без них.

Определение достаточности покрытия классами CWE базы NVD. Необходимо убедиться в достаточности покрытия классами CWE базы NVD. Для этого организуем поиск по ключевым словам в описании баз с постфиксом 2.0 и выборкой по ключевому слову «vuln:cwe» для предыдущей версии баз. Таким образом, получаем следующий график (рис. 1)

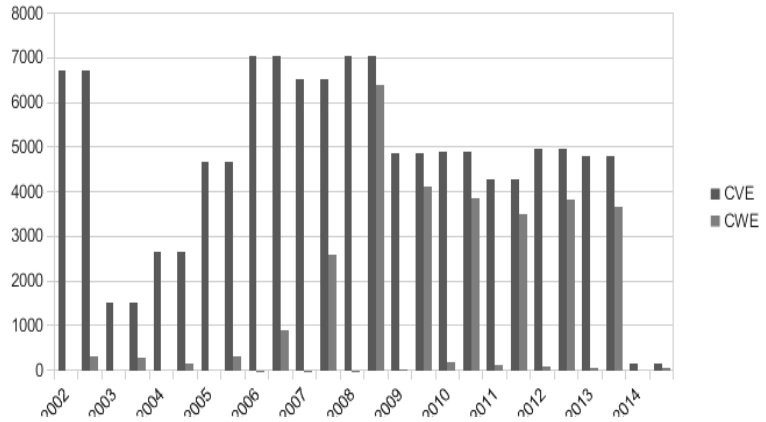

Рис. 1. Покрытие классами CWE базы NVD

Как видно на рис. 1, наиболее информативными по классам CWE являются базы NVD версии 2.0 (первый парный столбец в году). Соответственно для проведения исследований можно воспользоваться только ими, так как количество описанных уязвимостей в базе версии 2.0 и без данного постфикса – совпадают. Также базы с 2008 и по 2014 гг, в основном, обладают покрытием более 80 %, что является достаточным для построения карты уязвимостей. Таким образом, мы сокращаем время на анализ уязвимостей.

Для удобного представления и анализа уязвимости в программном обеспечении может быть построена карта уязвимостей, которая будет содержать данные из баз уязвимостей «National Vulnerability Database» (NVD) и классов уязвимостей «Common Weakness Enumeration» (CWE).

Построение карты уязвимостей. Карта уязвимостей отображает «покрытие» уязвимостями той или иной версии продукта и необходима для определения, «пересекаются» ли уязвимости разных версий одного программного продукта. Если это происходит, то можно сделать вывод, что уязвимости либо мутируют, либо остаются прежними, соответственно есть возможность воспользоваться эксплоитом для предыдущей версии или наоборот от новой версии к старой или же проэксплуатировать один программный компонент эксплоитом от другого, схожего по назначению компонента.

Выделение часто встречающихся уязвимостей. Исходя из данных, полученных из карты уязвимостей, возможно определить по классификации CWE, какие именно повторяются в разных версиях, таким образом определить мутацию.

Выделение часто встречающихся уязвимостей в случае больших программных систем может быть выполнено с использованием простейшего статистического анализа. На деталях подобного анализа в данной работе мы не останавливаемся.

Принципы выбора версий программного обеспечения для возможной эксплуатации. Эксплуатация может быть произведена не только на старшую версию, а также на продукты, использующие общий уязвимый компонент. Кроме того, уязвимость для новых версий продукта может быть актуальна и не исправлена в более ранних версиях. Так можно определить три ключевых направления атаки:

- «вверх» – атака, в которой предполагается сохранение уязвимости из младшей версии продукта в старшей;
- «вниз» – атака, в которой предполагается сохранение уязвимости из старшей версии продукта в младшей;

- «вдоль» – атака, в которой предполагается сохранение уязвимости в смежном программном обеспечении, использующих уязвимый программный модуль.

Экспериментальная оценка методики. Экспериментальная оценка методики производится на продукте MS Office как наиболее актуальном офисном приложении.

Карта уязвимостей для продукта «MS Office». Для построения карты был произведен анализ файлов, доступных на сегодняшний день, NVD версии 2.0.

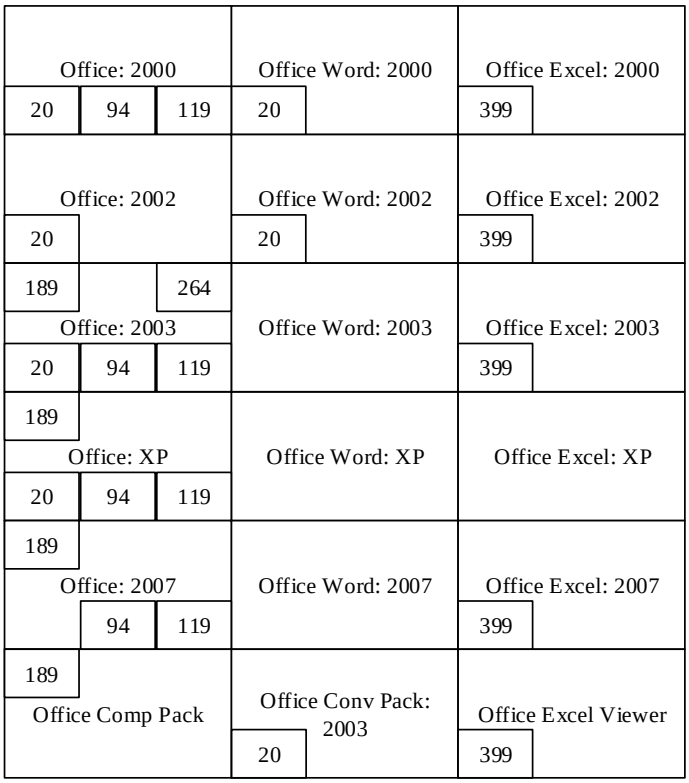

Рис. 2. Карта уязвимостей продукта MS Office

Как видно из карты уязвимостей (рис. 2), классы уязвимостей, в большинстве своем, повторяются во всех версиях от 2000 до версии 2008 г. Таким образом, можно сделать предположение, что цель уязвимостей осталась неизменна, потому можно провести эксплуатацию, используя модифицированные эксплоиты. Для этого необходимо выделить семейство эксплоитов.

Выделение классов наиболее повторяющихся уязвимостей. Как видно из рис. 2 для системы MS Office наиболее актуальны следующие уязвимости*:*

CWE-20 – некорректная обработка ввода. Позволяет осуществить атаку «Отказ в обслуживании» и вызвать аварийное завершение программы или перезапуск. [5].

CWE-94 – некорректный контроль генерируемого кода («Внедрение вредоносного кода»), может быть проэксплуатирована удалённо [5].

CWE-119 – некорректное ограничение операций в рамках буфера памяти. Атакующий может выполнить произвольный код, используя переполнение буфера.

CWE-189 – некорректный расчет и преобразование чисел [5].

CWE-264 – некорректное управление доступом и привилегиями. [5].

CWE-399 – некорректное управление ресурсами ОЗУ. Позволяет управлять системными ресурсами.

Исходя из собранной информации об уязвимостях, необходимо подобрать рабочий эксплоит для уязвимости, которая эксплуатировалась в ранней версии продукта (напр.: office:2000) и с помощью внесения небольших изменений (или без [6]) проэксплуатировать более новую версию (напр.: Microsoft:office:2008). Также исходя из предложенных критериев, необходимо найти эксплоит семейства CWE-399 для Microsoft Office. Выбор эксплоитов будет производиться путем поиска по базам такого продукта, как Metasploit Framework и иным публичным базам эксплоитов.

Атака «вверх». Для осуществления данного вида атаки необходимо подобрать эксплоит к семейству уязвимостей (CWE), содержащихся в карте уязвимостей, которая присутствует как в младшей версии продукта, так и в старшей. Так, например, можно использовать CWE-399, которая распространяется на версию Microsoft Office 2000 и Office 2007.

Таким образом, найдя подходящий эксплоит для семейства уязвимостей CWE-399, конкретно для версии Microsoft Office 2000 существует возможность эксплуатировать в неизменном виде или внести небольшие изменения, с помощью которых будет обнаружена неизвестная уязвимость, являющаяся мутацией уязвимости в предыдущей версии продукта.

В качестве экспериментального доказательства выступает пример эксплоита, использующий уязвимость с идентификатором CVE-2006-2389, который без каких-либо изменений мог эксплуатировать уязвимости в продуктах MS Office с версии 2000 до XP SP3, между которыми прошло 3 мажорные версии и 4 пакета обновлений [7]. Уязвимость позволяла злоумышленнику выполнить произвольный код в специально подготовленном файле формата Office. Исправления к данной уязвимости были так же опубликованы вначале для Office 2000 (KB923090 – 10/5/2006), а затем для Office XP (KB917150 – 12/12/2006)[8].

Атака «вниз». Для осуществления данного вида атаки необходимо подобрать эксплоит к семейству уязвимостей, которая присутствует в старшей версии продукта и распространяется на младшую. Так, например можно использовать эксплоит к семейству уязвимостей CWE-189, который распространяется как на версии Microsoft Office 2007, так и для атаки на Microsoft Office 2003. В случае неудачной эксплуатации уязвимостей без модификации эксплоита, проанализировав результат, следует внести небольшие изменения и, если эксплуатация будет успешна, в таком случае можно считать, что данная уязвимость также является мутировавшей.

В качестве экспериментального доказательства выступает уязвимость с идентификатором CVE-2006-3431, которая распространяется на продукты MS Office 2003, XP и 2000 версии. Данная уязвимость была обнаружена в 2006 г. для версии XP и позволяла злоумышленнику выполнить произвольный код после переполнения буфера, который вызывался специально измененным файлом в формате office, позже была подтверждена актуальность для более поздних версий [9].

Атака «вдоль». Следуя карте уязвимостей, можно обнаружить, что общие семейства уязвимостей распространяются не только на версии одного и того же продукта, но и на смежные программные продукты. Например, CWE-399 распространяется как на MS Office Excel, так и на MS Office Project, следовательно, можно предположить, что данная уязвимость будет актуальная для MS Word или иных смежных продуктов. Эксплоит для данной группы уязвимостей может быть подвергнут небольшим изменениям, что, в свою очередь, подтвердит мутацию данной уязвимости.

В качестве экспериментального доказательства выступает уязвимость «Microsoft Tagged Image File Format», которая распространяется как на продукты MS Office, так и компоненты операционной системы Windows, использующих общую библиотеку для отображения изображений в формате TIFF. Используя эту уязвимость, злоумышленник мог выполнить произвольный код от имени текущего пользователя.

Заключение. Идеи и методы, представленные в данной работе, служат цели обнаружения неизвестных уязвимостей на этапе разработки и тестирования программного обеспечения.

Представленные виды атак, полученные из карты уязвимостей, позволяют находить уязвимости в любой версии при наличии известных уязвимостей и эксплоитов к ним, по которым можно определить, как необходимо модифицировать имеющийся эксплоит для атаки на мутировавшую уязвимость. По каждому виду атак были получены экспериментальные результаты, которые отображают состоятельность разработанной методики к обнаружению уязвимостей в различных версиях ПО и смежном ПО, которое использует общие уязвимые библиотеки.

Кроме того, разработанная методика лишена недостатков динамического анализа, т.е. не требует высоких временных затрат и ресурсов, так как не нуждается в средствах виртуализации или анализа трассы выполнения программы.

В рамках дальнейшего развития данной работы предполагается выявление общих признаков мутации для построения единого алгоритма поиска для всех видов атак, а также использование анализа выпускаемых пакетов исправлений [10] как дополнительный источник данных для построения карты уязвимостей программного обеспечения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Selvaraj K., Gutierrez N.F.* The rise of PDF malware // Symantec Security Response. – 2010.
2. *Hueske F., Krettek A., Tzoumas K.* Enabling operator reordering in data flow programs through static code analysis // arXiv preprint arXiv:1301.4200. – 2013.
3. *Mallouli W. et al.* VDC-Based Dynamic Code Analysis: Application to C Programs // Journal of Internet Services and Information Security. – 2011. – Vol. 1, №. 2/3. – P. 4-20.
4. *Tzermias Z. et al.* Combining static and dynamic analysis for the detection of malicious documents // Proceedings of the Fourth European Workshop on System Security. – ACM, 2011. – P. 4.
5. MITRE Corporation, Common Weakness Enumeration, 2014. [Электронный ресурс]. –Режим доступа: http://cwe.mitre.org/, свободный.
6. SecurityLab, Microsoft не будет исправлять уязвимость в Internet Explorer 8 семимесячной давности, 2014. [Электронный ресурс]. – Режим доступа: http://www.securitylab.ru/news/453198.php, свободный.
7. MITRE Corporation, Common Weakness Enumeration. URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2389.
8. Microsoft, KB-917150, 2014. [Электронный ресурс]. – Режим доступа: http://support.microsoft.com/kb/917150/ru, свободный.
9. MITRE Corporation, Common Weakness Enumeration. URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3431.
10. *Arora A. et al.* An empirical analysis of software vendors' patch release behavior: impact of vulnerability disclosure // Information Systems Research. – 2010. – Vol. 21, №. 1. – P. 115-132.

REFERENCES

1. *Selvaraj K., Gutierrez N.F.* The rise of PDF malware, *Symantec Security Response*, 2010.
2. *Hueske F., Krettek A., Tzoumas K.* Enabling operator reordering in data flow programs through static code analysis, arXiv preprint arXiv:1301.4200, 2013.
3. *Mallouli W. et al.* VDC-Based Dynamic Code Analysis: Application to C Programs, *Journal of Internet Services and Information Security*, 2011, Vol. 1, No. 2/3, pp. 4-20.

4. *Tzermias Z. et al.* Combining static and dynamic analysis for the detection of malicious documents, Proceedings of the Fourth European Workshop on System Security. ACM, 2011, pp. 4.
5. MITRE Corporation, Common Weakness Enumeration, 2014. Available at: http://cwe.mitre.org/.
6. SecurityLab, Microsoft ne budet ispravlyat' uyazvimost' v Internet Explorer 8 semimesyachnoy davnosti, 2014 [SecurityLab, Microsoft will not fix the vulnerability in Internet Explorer 8 seven-month-old]. Available at: http://www.securitylab.ru/news/453198.php.
7. MITRE Corporation, Common Weakness Enumeration. Available at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2389.
8. Microsoft, KB-917150, 2014. Available at: http://support.microsoft.com/kb/917150/ru, свободный.
9. MITRE Corporation, Common Weakness Enumeration. Available at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3431.
10. *Arora A. et al.* An empirical analysis of software vendors' patch release behavior: impact of vulnerability disclosure, *Information Systems Research,* 2010, Vol. 21, No. 1, pp. 115-132.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

Катаргин Дмитрий Андреевич – Южный федеральный университет; e-mail: dakatargin@sfedu.ru; 346880, Ростовская обл., г. Батайск, Северный Массив, 15, кв. 51; тел.: +79286211661; аспирант.

Katargin Dmitry Andreevich – Southern Federal University; e-mail: dakatargin@sfedu.ru; 15, Severniy Massiv, kv. 51 Bataysk, Rostovskaya obl, 346880; phone: +79286211661; postgraduate student.

УДК 004.056

Я.В. Тарасов

МЕТОД ОБНАРУЖЕНИЯ НИЗКОИНТЕНСИВНЫХ DDOS-АТАК НА ОСНОВЕ ГИБРИДНОЙ НЕЙРОННОЙ СЕТИ

Представлены результаты разработки метода обнаружения сетевых атак типа «отказ в обслуживании» на различные сервисы хранения, обработки и передачи данных в сети Интернет. Основное внимание уделено обнаружению низкоинтенсивных атак типа «отказ в обслуживании». Опровергается мнение, что специальные средства для обнаружения атак «отказ в обслуживании» (denial of service, DoS) не требуются, поскольку факт DoS-атаки невозможно не заметить. Показано, что для эффективного противодействия необходимо знать тип, характер и другие показатели атаки «отказ в обслуживании», а системы обнаружения распределённых атак позволяют оперативно получить эти сведения. Кроме того, использование такого рода систем обнаружения атак позволяет существенно уменьшить время определения факта проведения атаки – с 2–3 суток до нескольких десятков минут, что снижает затраты на трафик и время простоя атакуемого ресурса. В качестве модуля обнаружения используется гибридная нейронная сеть на основе сети Кохонена и многослойного персептрона. Описана работа созданного прототипа системы обнаружения атак, методика формирования обучающей выборки, ход экспериментов и топология экспериментального стенда. Представлены результаты экспериментального исследования прототипа, в ходе которых ошибки первого и второго рода составили соответственно 3,16 и 1,23 %.

Обнаружение атак; низкоинтенсивные DDoS-атаки; гибридная нейронная сеть; безопасность вычислительных сетей.