УДК 658.012
С.В. Покрова, ассистент,
Д.О. Зганяйко, магистр
Таврический национальный университет им. В.И. Вернадского.
Просп. Вернадского 4, г. Симферополь, 95000.
E-mail: zdo.str@gmail.com

АВТОМАТИЗАЦИЯ ОБНАРУЖЕНИЯ И АНАЛИЗА РАЗЛИЧНЫХ КЛАССОВ УЯЗВИМОСТЕЙ ПРОГРАММНЫХ ПРОДУКТОВ

Рассмотрены базы данных уязвимостей программного обеспечения. Разработано программное решение для обработки исходных данных базы National Vulnerability Database. С помощью разработанного программного обеспечения выполнен анализ критериев CWE и CVSS, получена статистика по уязвимостям ряда программных продуктов, являющихся составными частями типовой архитектуры веб-сервисов.
***Ключевые слова**: уязвимость, база данных уязвимостей, CVE, дефект, CVSS.*

Введение. В течение последних лет наблюдается рост количества программных продуктов, решающих самые разные задачи в той или иной сфере человеческой деятельности. Хотя всего несколько десятков лет назад количество программного обеспечения было настолько мало, что зачастую людям, работающим с компьютерами, приходилось самим писать программы для решения своих собственных задач. Однако, сегодня количество ПО и разнообразие решаемых им задач так велико, что в большинстве случаев выбирается один из готовых программных продуктов, а не пишется новый с нуля, как это было не так давно.

С другой стороны, разнообразие программных средств не всегда хорошо сказывается на их качестве: постоянная конкуренция и жесткие условия рынка диктуют разработчикам свои правила создания ПО. Иногда в угоду более быстрому выпуску продукта или в силу каких-либо других причин на рынок выходит программное решение, которое оказалось недотестировано и ряд ошибок остался в релизной версии. Далее выпускаются патчи – обновления, в которых устраняются допущенные ошибки. Но за то время, пока использовалась неисправленная версия продукта, взломщики могли найти и эксплуатировать наличествующую уязвимость. И ситуация лишь усугубляется, если этот программный продукт – сетевой и хранит в себе конфиденциальную информацию своих пользователей, а искомая уязвимость – дефект в системе безопасности, позволяющий атакующей стороне сделать то, что по регламенту ей делать было бы не положено.

Как следствие, начали создаваться системы мониторинга уязвимостей программных продуктов. Одной из первых систем является NVD CVE [1], которая хранит в себе информацию о дефектах с 1999 года.

Цель. Стоит отметить, что сама по себе информация об уязвимостях не так важна, как своевременное реагирование на появление новых. И ныне отсутствует интегрированное программное решение, реализующее функционал быстрого уведомления пользователя об уязвимостях в интересующих его программных продуктах. В связи с этим была поставлена цель разработать прототип программной системы, который бы осуществлял сбор информации об уязвимостях, их анализ и классификацию по определенным критериям. Это позволит отыскать закономерности в допускаемых дефектах и выработать общие правила по минимизации количества подобных ошибок при разработке программного обеспечения.

Основываясь на понятиях, введенных разработчиками CVE, а именно классификаторах CVSS и CWE[2], была разработана программная система, выполняющая их автоматизированный анализ и статистическую обработку. Это набор взаимосвязанных скриптов на языке программирования Perl, использующих в качестве хранилища документо-ориентированную базу MongoDB.

Исходная база NVD содержит более 60000 уязвимостей программных решений самого разного класса. Для конкретизации анализа количество рассматриваемых уязвимостей было сужено до уязвимостей ряда программных продуктов, отражающих типовое ПО в структурной схеме веб-сервисов [3], а именно:

- **серверные операционные системы**: Microsoft Windows Server 2003, Server 2008, Server 2012, Linux (2.4-3.2);
- **клиентские операционные системы**: Microsoft Windows XP, Vista, 7, 8;
- **языки программировани**я: PHP, Perl;
- **веб-серверы**: Apache HTTP Server, nginx, Microsoft IIS;
- **СУБД**: MongoDB, Microsoft SQL Server 2000, 2005, 2008, 2012, MySQL, PostgreSQL.

Первым этапом стали операции парсинга, конвертации и унификации данных NVD, выполненные с помощью разработанной утилиты, общая блок-схема алгоритма которой изображена на рисунке 1.

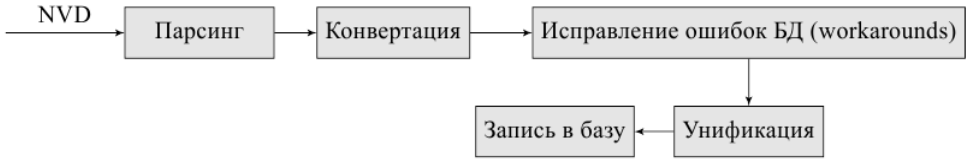

Рисунок 1 – Блок-схема работы алгоритма разработанной программы

Следующим шагом стал анализ по критерию CWE (Common Weakness Enumeration). Поскольку CWE является обширным и строго формализованным классификатором, содержащим более 1000 наименований различных типов уязвимостей, была разработана собственная классификация, представляющая собой кластеризацию отдельных наименований CWE в специальные классы, как показано в таблице 1.

Таблица 1 – Внутренняя классификация уязвимостей на базе CWE

Внутренний id	**CWE id**	**Описание**
stack-ou	CWE-134	Переполнение стека
buffer-ou	CWE-119	Переполнение буфера
inputvalid	CWE-20, CWE-79 CWE-78, CWE-89, CWE-94, CWE-22, CWE-59, CWE-352	Валидация входных данных, code injection (SQLi, CSS)
invptr	CWE-476	Разыменование некорректного (или нулевого) указателя
memfail	CWE-399	Утечки памяти
thread	CWE-362	Ошибки синхронизации и параллельного взаимодействия
numerr	CWE-189	Деление на ноль, целочисленное переполнение, точность чисел с плавающей запятой
auth	CWE-264, CWE-287, CWE-255	Недостатки систем аутентификации
debug	CWE-200, CWE-16	Отладочные сборки в релизном окружении

После проведения анализа указанного списка ПО по данной классификации были получены данные, указанные в таблице 2.

Как видно из полученных данных, бóльшая часть уязвимостей относится к классам «валидации входных данных» и «утечек памяти». Исходя из этого, можно предположить, что эти типы ошибок имеют наибольшее распространение, а следовательно, составляют наибольшую опасность.

На заключительном этапе был проведен анализ по критерию CVSS, основная задача которого — минимизировать субъективное влияние на параметры вносимых уязвимостей и формализовать их. В результате были проанализированы и получены следующие критерии:

• средние значения CVSS base score, по сути, показатели «критичности» уязвимости, значения которых указали на меньшее количество уязвимостей для проприетарных продуктов по сравнению с opensource (что, очевидно, связано с различной политикой обнародования уязвимостей);

• вектор уязвимости, среднее значение которого указывает на преобладающее количество сетевых уязвимостей по сравнению с эксплуатируемыми локально;

• критерий сложности эксплуатации, среднее значение которого показало, что большинство уязвимостей относятся к классу легкоэксплуатируемых.

Таблица 2 – Распределение уязвимостей исследуемых программных продуктов согласно внутренней классификации

Продукт	**Всего**	Без CWE	stack-ou	buffer-ou	inputvalid	invptr	memfail	thread	numerr	auth	debug
Windows Server 2003	335	**39** 11,64%	**1** 0,30%	**35** 10,45%	**109** 32,54%	**0** 0,00%	**52** 15,52%	**35** 10,45%	**20** 5,97%	**35** 10,45%	**7** 2,09%
Windows Server 2008	412	**48** 11,65%	**1** 0,24%	**39** 9,47%	**131** 31,80%	**0** 0,00%	**72** 17,48%	**39** 9,47%	**27** 6,55%	**43** 10,44%	**10** 2,43%
Windows Server 2012	24	**1** 4,17%	**0** 0,00%	**5** 20,83%	**4** 16,67%	**0** 0,00%	**2** 8,33%	**4** 16,67%	**3** 12,50%	**5** 20,83%	**0** 0,00%
Windows XP	767	**362** 47,20%	**1** 0,13%	**67** 8,74%	**140** 18,25%	**0** 0,00%	**63** 8,21%	**39** 5,08%	**31** 4,04%	**54** 7,04%	**7** 0,91%
Windows Vista	434	**66** 15,21%	**1** 0,23%	**45** 10,37%	**129** 29,72%	**0** 0,00%	**72** 16,59%	**42** 9,68%	**27** 6,22%	**42** 9,68%	**9** 2,07%
Windows 7	275	**42** 15,27%	**1** 0,36%	**21** 7,64%	**74** 26,91%	**0** 0,00%	**54** 19,64%	**37** 13,45%	**14** 5,09%	**28** 10,18%	**3** 1,09%
Windows 8	27	**2** 7,41%	**0** 0,00%	**5** 18,52%	**4** 14,81%	**0** 0,00%	**3** 11,11%	**4** 14,81%	**4** 14,81%	**5** 18,52%	**0** 0,00%
Linux	1156	**550** 47,58%	**2** 0,17%	**90** 7,79%	**85** 7,35%	**0** 0,00%	**129** 11,16%	**29** 2,51%	**78** 6,75%	**87** 7,53%	**97** 8,39%
Apache Web Server	235	**163** 69,36%	**0** 0,00%	**6** 2,55%	**26** 11,06%	**0** 0,00%	**15** 6,38%	**2** 0,85%	**7** 2,98%	**5** 2,13%	**9** 3,83%
Nginx	10	**0** 0,00%	**0** 0,00%	**4** 40,00%	**3** 30,00%	**0** 0,00%	**1** 10,00%	**0** 0,00%	**0** 0,00%	**1** 10,00%	**1** 10,00%
Microsoft IIS	173	**134** 77,46%	**0** 0,00%	**5** 2,89%	**10** 5,78%	**0** 0,00%	**1** 0,58%	**1** 0,58%	**1** 0,58%	**8** 4,62%	**11** 6,36%
Perl	14	**3** 21,43%	**1** 7,14%	**1** 7,14%	**1** 7,14%	**0** 0,00%	**2** 14,29%	**0** 0,00%	**3** 21,43%	**3** 21,43%	**0** 0,00%
PHP	370	**188** 50,81%	**6** 1,62%	**32** 8,65%	**56** 15,14%	**0** 0,00%	**18** 4,86%	**2** 0,54%	**23** 6,22%	**24** 6,49%	**17** 4,59%
MongoDB	1	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**1** 100,00%	**0** 0,00%
MSSQL 2000	53	**41** 77,36%	**0** 0,00%	**5** 9,43%	**2** 3,77%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**1** 1,89%	**3** 5,66%	**1** 1,89%
MSSQL 2005	23	**0** 0,00%	**0** 0,00%	**8** 34,78%	**5** 21,74%	**0** 0,00%	**1** 4,35%	**0** 0,00%	**7** 30,43%	**0** 0,00%	**2** 8,70%
MSSQL 2008	4	**0** 0,00%	**0** 0,00%	**0** 0,00%	**3** 75,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**1** 25,00%
MSSQL 2012	1	**0** 0,00%	**0** 0,00%	**0** 0,00%	**1** 100,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%	**0** 0,00%
MySQL	219	**171** 78,08%	**3** 1,37%	**9** 4,11%	**15** 6,85%	**0** 0,00%	**8** 3,65%	**0** 0,00%	**3** 1,37%	**6** 2,74%	**1** 0,46%
PostgreSQL	73	**38** 52,05%	**0** 0,00%	**2** 2,74%	**6** 8,22%	**0** 0,00%	**3** 4,11%	**0** 0,00%	**6** 8,22%	**15** 20,55%	**1** 1,37%

Продолжение исследований авторы видитят в дальнейшей разработке программного комплекса с целью полностью автоматизировать процесс исследования, а именно: объединение данных из различных баз (в том числе применение специализированных багтрекеров), регулярное обновление, построение динамики роста уязвимостей по различным критериям и генерация статистической информации по критериям, предлагаемыми различными базами данных уязвимостей.

Библиографический список использованной литературы

1. National Vulnerability Database [Электронный ресурс]. — Режим доступа: http://nvd.nist.gov

2. CWE Version 2.5. Common Weakness Enumeration – A Community-Developed Dictionary of Software Weakness Types [Электронный ресурс]. — Режим доступа: http://cwe.mitre.org/data/published/cwe_v2.5.pdf

3. Куланов С.А. Modeling of Dependable Systems and Networks — Моделирование гаранто-способности систем и сетей / С.А. Куланов, В.Н. Локазюк, под ред. Харченко В.С. — Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2008. — 335 с.

Поступила в редакцию 20.01.2014 г.

Покрова С.В., Зганяйко Д.О. Система автоматизації виявлення та аналізу різних класів уразливостей програмних продуктів

Розглянуті бази даних уразливостей програмного забезпечення. Розроблено програмне рішення для обробки вихідних даних бази National Vulnerability Database. За допомогою розробленого ПЗ виконаний аналіз критеріїв CWE та CVSS, отримана статистика щодо уразливостей ряду програмних продуктів, які є составною часткою типової архітектури веб-сервісів.

Ключові слова: уразливість, база даних уразливостей, CVE, дефект, CVSS.

Pokrova S.V., Zganyaiko D.O. The automatization system of detecting and analysis of different vulnerability classes in the set of common software products

Software vulnerability databases were examined. Application solution for National Vulnerability Database processing was developed. Using this software CWE and CVSS analysis was performed, statistics about vulnerabilities of set of products which are parts of typical web-service architecture were obtained.

Keywords: vulnerability, vulnerability database, CVE, defect, CVSS.