

БОЙКО ВИКТОР ДМИТРИЕВИЧ

*Национальный университет «Одесская юридическая академия»,
кандидат технических наук, доцент кафедры кибербезопасности*

ВОПРОСЫ ПЕРЕХОДА НА СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В СОВРЕМЕННОМ АКАДЕМИЧЕСКОМ ОБРАЗОВАНИИ

На заре внедрения программного обеспечения (ПО) в образовательный процесс была широко распространена практика использования пиратского (взломанного, либо не лицензированного явно) программного обеспечения. Одной из главных причин было то, что контроль за использованием лицензионно-чистого ПО не был проработан ни законодательно, ни технически.

В настоящее время контроль за использованием ПО усиливается, что приводит к необходимости лицензирования существующего ПО, либо к переходу на открытое ПО, что многими воспринимается, как вынужденная мера. Такой выбор представляет собой принципиальный процесс, поскольку два направления разработки представляют собой два разных подхода – открытый, при котором клиенту поставляются в том числе исходные коды программного обеспечения и закрытый – когда клиенту поставляются только бинарные файлы.

Среди разработчиков и профессионалов в IT-отрасли сложился устойчивый тренд к увеличению и расширению использования средств открытого ПО. Например, согласно опросу Forrester Research от 2014-го года [1] 84 % разработчиков (в том числе из компаний с проприетарным ПО) используют программное обеспечение с открытым исходным кодом. При этом большая часть разработчиков использует ПО исходя из его производительности и надёжности, в большинстве случаев стоимость играет для них последнюю очередь. Аналогичные тренды сложились в науке и образовании [2; 3].

Не в последнюю очередь это происходит постольку поскольку проприетарные, закрытые исходные коды, протоколы взаимодействия и форматы данных не смотря на свою привлекательность – «решим все ваши проблемы» (разработчик берет на себя поставку и развертывание системы «под ключ», а также обеспечивает дальнейшую техническую поддержку), на практике порождают множество проблем [4; 5].

В их числе: «vendor lock-in» – стремление поставщиков услуг замкнуть клиента на себя, монополизировав оказание услуги, возможное наличие «черных ходов» («backdoors»), неоправданное раздувание объемов ПО и повышение требований к используемому парку аппаратного обеспечения и так далее. Кроме того, при использовании закрытого программного обеспечения, возникает риск потери возможности обслуживания системы в том случае, если ее разработчик обанкротится, уйдет с рынка или его компания будет поглощена другим разработчиком.

Все это служит существенным аргументом в пользу внедрения открытого ПО в образовательный процесс.

Опыт перевода рабочих процессов на открытое ПО позволил выработать несколько практических рекомендаций. При этом основным принципом должна быть постепенность и посильность процесса перехода. Любая смена программного обеспечения неизбежно связана со сменой рабочих процессов, а значит – с «детскими болезнями», с подбором новых инструментов и – что особенно важно – со сменой привычек и организации рабочих процессов, поскольку человеческий фактор составляет существенную часть. Наихудшей является ситуация «нужно было вчера».

Для обеспечения постепенности внедрения можно придерживаться следующих рекомендаций.

- проанализируйте какие Open Source средства и открытые форматы данных вы можете использовать уже сейчас – под текущей операционной системой

Существенная часть открытого ПО кроссплатформенна – то есть существует в разных версиях для разных операционных систем и архитектур. Поэтому процесс внедрения, обучения и использования открытого ПО можно начинать даже не меняя платформу, а постепенно меняя пользовательское ПО. Например, сменить почтовый клиент на Mozilla Thunderbird, в качестве офисного пакета использовать Libre Office и т. д. и т. п.

- изучите отличия платформы на которую собираетесь перейти

У разных операционных систем зачастую гораздо больше общего, чем разного, поскольку существует своеобразный аналог биологической конвергенции для компьютерных систем. Однако, некоторые существующие различия могут сбивать с толку, затрудняя переход. К этому следует быть готовым.

- начните переход с использования открытого ПО, открытых форматов данных и выстраивания рабочих процессов на текущей рабочей платформе

Как было сказано выше, это вполне осуществимо в большинстве случаев.

- проведите постепенный переход сохраняя возможность возврата к платформе

По нашему опыту, наиболее оптимальная схема – это внедрение пилотного проекта, для постепенного перевода пользователей на новую систему. Как было сказано выше, следует быть готовым к «детским болезням» внедрения – и схема с пилотным проектом позволяет решить эти проблемы «малой кровью» без существенных потерь, связанных с неожиданными трудностями внедрения.

Обычно внедрение начинается с одной-двух аудиторий, которые служат полигоном для выявления всех ошибок и проблем внедрения. По мере освоения системы, «поверхность внедрения» расширяется, занимая все больше классов. При этом на каждом этапе сохраняется

возможность возврата на предыдущий этап, если в этом возникнет необходимость.

- откажитесь от перфекционизма

Иногда в проектах может требоваться наличие старых систем. В этом случае, разумно будет оставить несколько старых участков до тех пор пока дальнейший рост освоения новых систем не позволит заменить их, либо перейти на использование виртуальных окружений – словом каким-либо образом решить проблему.

- продолжайте изучать новые средства новой платформы

Выбор схожего ПО и схем работы облегчает смену систем и переходный период, однако, часто (и это характерно для открытого ПО) можно значительно повысить КПД использования ПО, если задействовать сильные и уникальные стороны существующих систем. Например, LibreOffice, который обычно используется в режиме «бесплатный Microsoft Word» содержит в себе такие дополнительные инструменты, как удобный навигатор по документу, html/css-подобную систему задания стилей документа, составные документы (работа осуществляется через мастер-документ, который содержит в себе ссылки на другие документы), систему полей, систему библиографии, генератор содержания документа на основе стилиевой системы, систему текстового задания математических формул, схожую с LaTeX и так далее. Все это, наряду со встроенным экспортом в pdf-файлы делает LibreOffice удобным и мощным инструментом, а не только «бесплатным вариантом текстового процессора» [6; 7].

Однако, «глубокое освоение» тоже должно быть постепенным и посильным. К нему следует переходить только после завершения перехода и вывода рабочих процессов в оптимальный режим.

Список использованной литературы:

1. Survey indicates four out of five developers now use open source | ZDNet [Электронный ресурс]. – Режим доступа: <https://www.zdnet.com/article/survey-indicates-four-out-of-five-developers-now-use-open-source/> (дата звонения: 2021-05-04). – Назва з екрана.
2. Wiley D. Open source, openness, and higher education // Innovate: Journal of Online Education. – 2006. – Т. 3. – № 1.
3. Courant P. N., Griffiths R. J. Software and collaboration in higher education: A study of open source software // New York: Ithaca. Retrieved January. – 2006. – Т. 30. – С. 2009.
4. Ruffin C., Ebert C. Using open source software in product development: A primer // IEEE software. – 2004. – Т. 21. – № 1. – С. 82–86.
5. Bouras C., Kokkinos V., Tseliou G. Methodology for Public Administrators for selecting between open source and proprietary software // Telematics and Informatics. – 2013. – Т. 30. – № 2. – С. 100–110.
6. Gamalielsson J., Lundell B. Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved? // Journal of Systems and Software. – 2014. – Т. 89. – С. 128–145.

7. Llerena L. et al. Applying a Usability Technique in the LibreOffice Writer Project // CLEI ELECTRONIC JOURNAL. – 2020. – Т. 23. – № 2.

Ключові слова: відкрите програмне забезпечення, відкриті формати даних, відкриті протоколи взаємодії, загрози і ризики, освіта.

Ключевые слова: открытое программное обеспечение, открытые форматы данных, открытые протоколы взаимодействия, угрозы и риски, образование.

Key words: open software, open data formats, open communication protocols, threats and risks, education.

СОКОЛОВ АРТЕМ ВИКТОРОВИЧ

*Национальный университет «Одесская юридическая академия»,
доцент кафедры кибербезопасности,
кандидат технических наук, доцент*

ХИМЕНКО МИХАИЛ ВАЛЕРЬЕВИЧ

*Государственный университет «Одесская политехника»,
студент бакалавриата*

ТЕСТИРОВАНИЕ ГЕНЕРАТОРОВ КЛЮЧЕВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ СОВЕРШЕННЫХ АЛГЕБРАИЧЕСКИХ КОНСТРУКЦИЙ

Одной из важнейших составных частей современных систем защиты информации являются криптографически стойкие генераторы псевдослучайных ключевых последовательностей (ГПКП). Важным свойством современных ГПКП является их способность вырабатывать псевдослучайные последовательности G большой длины $|G|$ на основе криптографического ключа K незначительной длины $|K|$, т. е. $|G| \gg |K|$. При этом, с одной стороны результирующая последовательность должна обладать значительным уровнем стохастического качества, тогда как с другой стороны её восстановление должно быть возможно только в случае наличия криптографического ключа K .

ГПКП нашли свои значительные применения в системах защиты информации, основанных на концепции оперативной смены ключа [1], для генерации векторов инициализации, являются основным элементом современных поточных шифров, обеспечивающим выработку гаммы, которая впоследствии накладывается на шифруемую информацию.

В настоящее время предложены также схемы ускорения шифрования информации блочными симметричными шифрами [2], основанные на использовании ГПКП. При этом качество шифрования информации будет напрямую зависеть от качества используемой гаммы.