Смирнов Семен Александрович – Сибирский университет науки и технологий; e-mail: simon.simriel@gmail.com; г. Красноярск, Россия; тел.: 89130407272; кафедра безопасности информационных технологий; аспирант.

Паротькин Николай Юрьевич – e-mail: nyparotkin@yandex.ru; тел.: 89509920640; кафедра безопасности информационных технологий; к.т.н.; доцент.

Золотарев Вячеслав Владимирович – e-mail: zolotorev@sibsau.ru; тел.: 89059738091; кафедра безопасности информационных технологий; зав. кафедрой; к.т.н.

Smirnov Simon Aleksandrovich – Siberian University of Science and Technology; e-mail: simon.simriel@gmail.com; Krasnoyarsk, Russia; phone: +79130407272; the Department of Information Technology Security; post graduate student.

Parotkin Nikolay Yuryevich – e-mail: nyparotkin@yandex.ru; phone: +79509920640; the Department of Information Technology Security; cand. of eng. sc.; associate professor.

Zolotarev Vyacheslav Vladimirovich – e-mail: zolotorev@sibsau.ru; phone: +79059738091; the Department of Information Technology Security; head of department; cand. of eng. sc.

УДК 004.056 DOI 10.18522/2311-3103-2024-6-218-229

В.Г. Жуков, С.В. Селигеев

ГОСУДАРСТВЕННОЕ РЕГУЛИРОВАНИЕ ИМЕНОВАНИЯ И ИДЕНТИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРОЦЕССАХ УПРАВЛЕНИЯ УЯЗВИМОСТЯМИ

Управление ИТ активами является базисом для построения эффективного процесса управления уязвимостями. Не имея представления о контролируемых ИТ активах, технически невозможно начать построение процесса управления уязвимостями. При наличии действующего процесса управления ИТ активами, одной из задач, являющейся важной для управления уязвимостями, является однозначное именование программного обеспечения как актива. Такое однозначное именование позволяет идентифицировать программное обеспечение и его уязвимости без применения активного сканирования узлов ИТ инфраструктуры, а только при помощи взаимодействия с системой управления ИТ активами. Технически данный подход можно назвать «пассивное обнаружение уязвимостей», однако использование существующих систем именования для его реализации является крайне трудоемкой задачей. Для того чтобы сделать пассивное обнаружение реальным, в работе авторами предлагается создать общий фундамент путем формирования концептуальной схемы и последующего создания системы стандартизированного именования и идентификации программного обеспечения, регулирование которой будет происходить централизованно на государственном уровне. В рамках рассмотрения существующих систем именования программного обеспечения, внимание уделяется проблемам CPE как со стороны специалистов на местах, а именно получение CPE-идентификаторов и трансляция информации о программном обеспечении в CPE-идентификатор, так и со стороны агрегатора данных об уязвимостях, а именно получение сведений об уязвимостях через CPE-идентификатор. Обнаруженные в ходе исследования проблемы применения CPE, а также проблемы взаимодействия с агрегаторами данных об уязвимостях из недружественных стран формируют предпосылки к формированию национальной системы государственного регулирования именования и идентификации программного обеспечения, которая устранит проблемы существующих систем именования программного обеспечения. В заключении приводятся преимущества национальной системы именования и идентификации программного обеспечения в случае ее создания и использования в реальных условиях всеми участниками процесса управления уязвимостями.

Информационная безопасность; управление уязвимостями; управление активами; CPE.

V.G. Zhukov, S.V. Seligeev

STATE REGULATION OF NAMING AND SOFTWARE IDENTIFICATION IN VULNERABILITY MANAGEMENT PROCESSES

IT asset management is the foundation for building an effective vulnerability management process. Without an understanding of the IT assets under control, it is technically impossible to start building a vulnerability management process. With an existing IT asset management process in place, one of the tasks that is essential to vulnerability management is to uniquely name software as an asset. This unambiguous naming allows the software and its vulnerabilities to be identified without actively scanning IT infrastructure nodes, but only by interacting with the IT asset management system. Technically, this approach can be called "passive vulnerability detection," but it is extremely labor-intensive to implement using existing naming systems. In order to make the possibility of passive detection more realistic, the authors propose to create a common foundation by forming a conceptual scheme and then creating a system of standardized naming and identification of software, the regulation of which will be centralized at the state level. As part of the review of existing software naming systems, attention is paid to CPE problems both on the part of on-site specialists, namely obtaining CPE identifiers and translating software information into a CPE identifier, and on the part of a vulnerability data aggregator, namely obtaining vulnerability information via a CPE identifier. The problems of CPE application, as well as the problems of interaction with vulnerability data aggregators from unfriendly countries, discovered in the course of the research form the prerequisites for the formation of a national system for state regulation of software naming and identification, which will eliminate the problems of existing software naming systems. In conclusion, advantages of the national system of software naming and identification are given in case of its creation and use in real conditions by all participants of the vulnerability management process.

Information security; vulnerability management; asset management; CPE.

Введение. В рамках организации работ по управлению уязвимостями программных и программно-аппаратных средств в целевой ИТ инфраструктуре в соответствии с требованиями нормативно-правовых актов, методических документов [1] и лучших практик [2] необходимо изначально идентифицировать ИТ активы, которые необходимо контролировать. Данное мероприятие по идентификации должно носить не ручной разовый характер, а иметь системное решение с применением специализированного программного обеспечения – системы управления ИТ активами. Система управления ИТ активами должна предоставлять не только информацию о физических и виртуальных узлах защищаемой ИТ инфраструктуры, но и актуальные детализированные сведения об установленном на этих узлах системном и прикладном программном обеспечении. Сам факт существования и оперативный доступ к актуальной непротиворечивой детальной информации об ИТ активах имеет ценность в контексте реализации эффективного процесса управления уязвимостями и в обеспечении информационной безопасности в целом. Так, например, потенциально существует возможность быстро обнаруживать некоторые уязвимости основываясь только лишь на информации из системы управления ИТ активами, без непосредственного сканирования самих целевых узлов ИТ инфраструктуры, и, как следствие, без создания вычислительной нагрузки на эти узлы и вычислительную сеть для осуществления удаленной коммуникации с ними.

Необходимым условием пассивного обнаружения уязвимостей является гарантия однозначной идентификации установленного программного обеспечения через уникальный стандартизированный идентификатор программного обеспечения, который хранится как в описании самой уязвимости, так и в системе управления ИТ активами. В качестве примера подобной системы именования и идентификации активов можно привести Common Platform Enumeration (CPE) и другие частные решения, которые есть у каждого производителя решений класса Vulnerability Management (VM) и других средств защиты информации. Качество решения задачи именования и идентификации программного обеспечения [3–6] определяет скорость и точность обнаружения уязвимостей, требуемые для этого вычислительные ресурсы. Отдельно стоит обратить внимание на проблему большого количества обнаруживаемых уязвимостей, так, например, рост количества об-

наруживаемых уязвимостей [7] и переход National Vulnerability Database (NVD) на новый формат данных CVE – JSON – привел к значительным задержкам в обработке уязвимостей [8] при их регистрации, из-за которых на момент мая 2024 года была обработана лишь 1/3 уведомлений об уязвимостях, полученных за начало года, т.е. 2/3 обнаруженных уязвимостей находятся в очереди на регистрацию и не имеют идентификатора уязвимости, связанного с уязвимым программным обеспечением. При этом, с учетом растущего ландшафта угроз [9], злоумышленники активно эксплуатируют некоторые из этих уязвимостей. Частично данная проблема может быть решена дорогостоящими средствами защиты информации, производители которых самостоятельно обеспечивают высокое качество экспертизы уязвимостей и автоматизацию их обнаружения в целевой ИТ инфраструктуре, но не все организации могут позволить себе их приобретение. Пассивное обнаружение уязвимостей через однозначную идентификацию установленного программного обеспечения позволит значительно уменьшить расходы и повысить оперативность обнаружения, снизит потребность в частоте и объеме активного обнаружения (сканирования), и, в целом, приведет к повышению эффективности процесса управления уязвимостями.

Таким образом, постоянное совершенствование существующих и создание новых решений задачи именования и идентификации программного обеспечения является актуальной научно-технической проблемой и требует проведения исследований и разработки системного решения для всех участников процесса управления уязвимостями.

В работе авторы рассматривают теоретические и практические аспекты применения существующих систем однозначного именования и идентификации программного обеспечения, а также существующие проблемы их применения. На основании анализа проблем вводится предложение о необходимости разработки национальной системы для государственного (обязательного и централизованного) регулирования именования и идентификации программного обеспечения, как минимум, в управлении уязвимостями. В качестве основы для системы предполагается использовать уже существующие Реестр российского программного обеспечения [10] и банк данных угроз ФСТЭК [11], как национальный агрегатор данных об уязвимостях программных и программно-аппаратных средств.

1. Существующие подходы и проблемы их применения. Гипотеза о возможности применения систем именования и идентификации программного обеспечения для проведения пассивного анализа (без непосредственного сканирования) на предмет наличия уязвимостей программного обеспечения рассматривалась ранее авторами в работе [12]. Наиболее перспективными для применения в рамках решаемой задачи, по мнению авторов, являются такие системы как: software identification (SWID), package URL (PURL), Common Platform Enumeration (CPE).

SWID состоит из структурированного набора элементов данных (формат XML), которые идентифицируют конкретное программное обеспечение, характеризуют его версию, определяют организации и отдельные лица, которые сыграли роль в его разработке и распространении, перечисляют используемые зависимости, устанавливают отношения между различным программным обеспечением, и предоставляют другие метаданные. Данная система является частью спецификации Security Content Automation Protocol (SCAP), что дает совместимость с Common Vulnerabilities and Exposures (CVE).

Сложность, применяя данной системы заключается в том, что на ее основе строится отдельный процесс управления программным обеспечением, основанным на установке SWID меток в соответствии со стандартом ISO/IEC 19770-2:2015. Данная метка устанавливается на конечном этапе процесса установки программного обеспечения. В рамках данной системы однозначное именование программного обеспечения является только частной задачей, а не основной целью, что делает ее не актуальной для рассмотрения в рамках данной работы. Так же необходимо учитывать тот факт, что в рамках организаций на территории Российской Федерации система не получила широкого применения.

Package URL (PURL) – это попытка стандартизировать существующие подходы к идентификации пакетов программного обеспечения. Особенность данного стандарта заключается в том, что помимо метаданных о пакете, описание содержит действительный адрес, по которому данный пакет можно получить. Запись идентификатора, согласно этому стандарту, представляет собой строку, оформленную в соответствии с заданным шаблоном, каждый элемент которого отражает конкретное значение.

PURL в сравнении с ранее упомянутым SWID, основной целью которого является однозначное именование программного обеспечения, является более релевантным в контексте пассивного анализа на наличие уязвимостей. Так же тот факт, что система направлена на именование программного обеспечения для Unix-подобных систем, делает ее актуальной в связи с массовым импортозамещением операционных систем на отечественные аналоги, такие как Astra Linux, РЭД ОС, ОС Альт.

Вышеописанные достоинства PURL перекрываются тем фактом, что записи в формате Package URL не совместимы с существующей системой правил описания уязвимостей CVE и отечественными ГОСТами: ГОСТ Р 56545-2015 и ГОСТ Р 56546-2015, т.е. для применения этой системы необходима интерпретация (преобразование, отображение) идентификатора PURL в формат, применяемый в рамках существующей системы описания уязвимостей – это делает систему такой же не применимой на данный момент времени, как и вышеописанный SWID.

CPE так же, как и SWID является частью спецификации SCAP и на данный момент применяется в рамках отечественных процессов управления уязвимостями. Основная идея этой системы заключается в том, что данные имена интегрируются в CVE, для описания конкретного программного обеспечения, которое подвержено уязвимости, а также его версии и установленных обновлений. Фактически имя (идентификатор) по CPE представляет из себя строку с разделителями, где каждое значение описывает отдельный атрибут программного обеспечения.

Запись CPE формируется (или выбирается) в момент регистрации CVE уязвимости, и в последствии попадает в словарь CPE, который доступен как в виде файла (периодически обновляемого), так и посредством API, что расширяет возможные границы применения данной системы именования.

NVD как и другие агрегаторы данных об уязвимостях, например CISA Known Exploited Vulnerabilities (KEV) Catalog является сервисом, но в связи с геополитической обстановкой в мире крайне высок шанс того, что сервисы, чья работа обеспечивается недружественными странами, либо будут заблокированы, либо прекратят предоставлять услуги организациям с территории Российской Федерации. Ярким примером подобных действий является кратковременное прекращение обслуживания сервисом Docker Hub Российской Федерации и ряда других стран. NIST, и, в частности, NVD подчинены недружественному государству, что так же увеличивает вероятность ограничения обслуживания. Более серьезной проблемой является предоставление этими сервисами неполной или заведомо ложной информации организациям из нашей страны или стран-союзников в деструктивных целях – саботаж работы сканеров безопасности/уязвимостей/анализа защищенности и систем класса VM, кибердиверсии.

Вышеописанная проблема является основной, но не единственной предпосылкой к созданию национальной системы для государственного регулирования именования и идентификации программного обеспечения, с целью обеспечения независимости от внешних источников. Второй и не менее важной предпосылкой являются проблемы практического применения данной системы в рамках процессов управления уязвимостями, что более подробно рассматривается в следующем разделе.

2. Анализ практической применимости CPE. Для начала стоит отразить основную идею применения CPE-идентификаторов, которая ранее рассматривалась исследователями в работе [12]. Каждый идентификатор CVE, будь то в NVD или других базах данных сканеров безопасности, содержит в себе, помимо описания основных свойств уязвимости и оценки ее опасности, CPE-идентификатор который отражает подверженное уязвимости конкретное программное обеспечение с указанием его версии. При наличии в системе управления ИТ активами описания программного обеспечения в виде CPE-идентификатора открывается возможность обнаружения уязвимых активов с использованием лишь данных из системы управления ИТ активами, то есть без непосредственного сканирования. Такое обнаружение, в отличие от активного [13], может значительно снизить ресурсы и время, затраченное на идентификацию уязвимых активов в ИТ инфраструктуре, что увеличит эффективность процесса управления уязвимостями.

Ранее, в [12] авторами уже описывались проблемы неполных CPE-идентификаторов и ограниченности CPE-словаря в следствии того, что записи в нем появляются в массе своей только в момент регистрации уязвимостей в NVD. Фактически записей для программного обеспечения, в котором на данный момент не обнаружены уязвимости нет, либо в описании уязвимости используется обобщённый (неполный) CPE-идентификатор, что ограничивает появление новых CPE-идентификаторов. Далее рассматривается применимость CPE с учетом использования реальных инструментов с целью практического выявления проблем отличных от вышеописанных.

В рамках проводимого авторами исследования в качестве системы управления ИТ активами использовался GLPI с плагином Fields для возможности добавления дополнительных полей, в сущности, описываемые GLPI, а именно добавление поля CPE в сущность программное обеспечение. Хотя GLPI и не является полнофункциональной системой управления ИТ активами ее функционала, а именно хранение сведений об активах, достаточно в рамках проводимого исследования. В качестве сборщиков данных об установленном на тестовых хостах программном обеспечении рассматривались сканер безопасности RedCheck, как специализированный инструмент для обнаружения уязвимостей, а также плагин для GLPI – FusionInventory, как инструмент для инвентаризации конечных устройств (АРМы, сетевое оборудование и т.д.).

Идея исследования заключается в формировании CPE-идентификаторов, либо их получения через API NVD на базе информации, полученной при помощи инструментов сбора данных, для их последующего применения в рамках пассивного обнаружения уязвимостей в ИТ инфраструктуре.

Рассмотрим какие данные о программном обеспечении можно получить при помощи инструментов, приведенных выше. Для сравнения FusionInventory был запущен в штатном режиме, RedCheck в режиме «Инвентаризация» с указанием дополнительной опции – OVAL-инвентаризация.

Не смотря на разные цели инструментов каждый выдал в результате работы схожий набор данных, из которых наиболее важными для формирования CPE-идентификатора являются имя и версия программного обеспечения. Однако RedCheck в силу своей изначальной специализированной направленности дает больше информации за счет дополнительной OVAL-инвентаризации, а именно обнаружения установленного ПО при помощи OVAL-определений из OVALdb [14]. Результаты OVAL-инвентаризации представляют из себя список установленного программного обеспечения в виде CPE-идентификаторов, однако в большей массе они неполные и не являются информативными для процесса управления уязвимостями. Так же OVAL-инвентаризация покрывает лишь малую часть программного обеспечения, установленного на сканируемом сетевом узле ИТ инфраструктуры, что так же понижает информативность.

Дополнительно RedCheck имеет возможность получения CPE-идентификатора посредством проведения сканирования хостов на уязвимости в режимах «Аудит» и «Пентест». В таком случае можно провести обратную интерпретацию CPE-идентификатора посредством API NVD и внести его в карточку целевого программного обеспечения в системе управления ИТ активами, но эти записи будут актуальны только до момента устранения уязвимости посредством обновления (изменится версия), что сделает CPE-идентификатор автоматически устаревшим. Последующие сканирования в следствие отсутствия уязвимостей не сгенерируют актуальный CPE-идентификатор, что делает применение сканирования на уязвимости как инструмента для получения данных о программном обеспечении не применимым.

Таким образом для формирования и поддержания в актуальном состоянии CPE-идентификаторов внутри системы управления ИТ активами возможно использовать лишь данные, полученные в результате инвентаризации.

Так как для получения CPE-идентификатора требуется название программного обеспечения в качестве примера рассмотрим СУБД PostgreSQL 16 на Windows-хосте и пакет python3 на хосте под управлением Linux.

По логике одним из вариантов автоматизации процесса является выделение имени и версии программного обеспечения из системы управления ИТ активами, с последующей подготовкой запроса к API NVD. Однако в рамах проводимого анализа удалось обнаружить ряд проблем, которые делают данную задачу неразрешимой.

Первой и наиболее значимой проблемой является то, что получаемые в результате инвентаризации имена программного обеспечения очень редко могут дать однозначный ответ при запросе к API NVD (для российских IP это может делаться и намеренно). Так, например, CPE-идентификатор для приведенного в пример СУБД PostgreSQL 16 будет выглядеть следующим образом «cpe:2.3:a:postgresql:postgresql:16.3:*:*:*:*:*:*:*». Однако при выполнении запроса к API с указанием поиска по ключевым словам – в данном случае поиск по непосредственному имени программного обеспечения PostgreSQL 16 – результат его выполнения содержит в себе 22 CPE-идентификатора. Данный результат можно объяснить тем, что СУБД PostgreSQL 16 имеет релизы не только для операционных систем семейства Windows, но и для различных дистрибутивов Linux, что порождает отдельные CPE-идентификаторы, их количество так же умножает и наличие различных патчей (обновлений) у версии 16. Приведенный результат является наиболее качественным, но может произойти и обратная ситуация – прямой поиск по имени Notepad++ (64-bit x64) приведет к нулевому результату, хотя CPE-идентификатор для данного программного обеспечения существует. Такая проблема характерна для большого количества программных продуктов в рамках операционной системы Windows, так как сведения о названии и версии технически попадают в систему от разработчиков, у каждого из которых свое видение того какую информацию необходимо отразить в этих полях.

Рассматривая операционные системы семейства Linux, в которых в сравнении с Windows более унифицированная система именования программного обеспечения, а именно расширенное наименование пакета отражающее название и версию любого установленного пакета, можно так же явно увидеть проблемы с применением CPE. Так в случае пакета python3, для которого CPE-идентификатор имеет вид «cpe:2.3:a:python:python:3.1:*:*:*:*:*:*:*», при запросе с прямым указанием имени получим лишь один CPE-идентификатор, который описывает расширение для поддержки RestfulAPI – «cpe:2.3:a:python3-restfulapi_project:python3-restfulapi:-:*:*:*:*:*:*:*». Так же в рамках отдельных дистрибутивов существует практика закрепления уязвимостей конкретного пакета за идентификатором операционной системы, в качестве примера можно привести уязвимость CVE-2018-14574 в программном обеспечении Jango, но при этом имеющая CPE-идентификатор «cpe:2.3:o:canonical:ubuntu_linux:18.04:*:*:*:*:*:*:*».

Потенциально существует возможность получения валидных CPE-идентификаторов посредством обращения к API NVD, при условии предварительной обработки имен программного обеспечения полученных в результате сбора данных, например посредством парсинга с помощью регулярных выражений. Однако из-за высокой степени уникальности получаемых имён программного обеспечения, данная задача является крайне трудоемкой, если вообще выполнимой. Сложность этой задачи помимо того, что ограничивает взаимодействие с API NVD, так же затрудняет второй вариант получения CPE-идентификаторов из собранных данных, а именно самостоятельную генерацию. Однако, не смотря на сложность, исследователями постоянно предпринимаются попытки автоматизации формирования CPE-идентификаторов при регистрации новых уязвимостей при помощи различных методов [15–17].

Отдельно стоит сказать про программное обеспечение отечественной разработки, которое повсеместно применяется в российских организациях [18–20]. Множество уязвимостей такого программного обеспечения имеют лишь идентификатор БДУ банка данных угроз ФСТЭК [11] и не содержат информацию о наличии CPE. В качестве примера такой уязвимости можно привести BDU:2021-05103 операционной системы Синергия.

Вышеописанные проблемы обосновывают необходимость создания национальной системы для государственного регулирования именования и идентификации программного обеспечения. Хоть некоторые инструменты и используют CPE-идентификаторы для обогащения описания обнаруженных уязвимостей, их применение в рамках изначальной идеи, заложенной в систему именования CPE в нынешних условиях, является кране сложной задачей.

3. Концептуальная схема национальной системы для государственного регулирования именования и идентификации программного обеспечения в Российской Федерации. Реестр российского программного обеспечения [10] взаимодействует с БДУ ФСТЭК [11] на уровне информирования пользователя о наличии уязвимостей в зарегистрированном программном обеспечении. В рамках реализации Национальной системы для государственного регулирования именования и идентификации программного обеспечения (СИИПО) предлагается расширение взаимодействия этих агрегаторов данных. Реестр отечественного программного обеспечения [10] предоставит техническую информацию для формирования уникальных идентификаторов программного и программно-аппаратного обеспечения, например, информацию о производителе, типе (классификаторы), имени программного обеспечения и иную информацию. Предположительная структура хранения данных о программном обеспечении представлена на рис. 1.

При регистрации новой уязвимости в БДУ ФСТЭК [11] предоставляется возможность выбора значений параметров уязвимого программного обеспечения из доступного фиксированного множества значений, определенных СИИПО, например, «операционная система» и «аппаратная платформа». В случае создания новой записи, например, регистрация первой уязвимости для новой версии программного обеспечения, заявки модерируются регулятором с целью контроля создания записей в системе хранения.

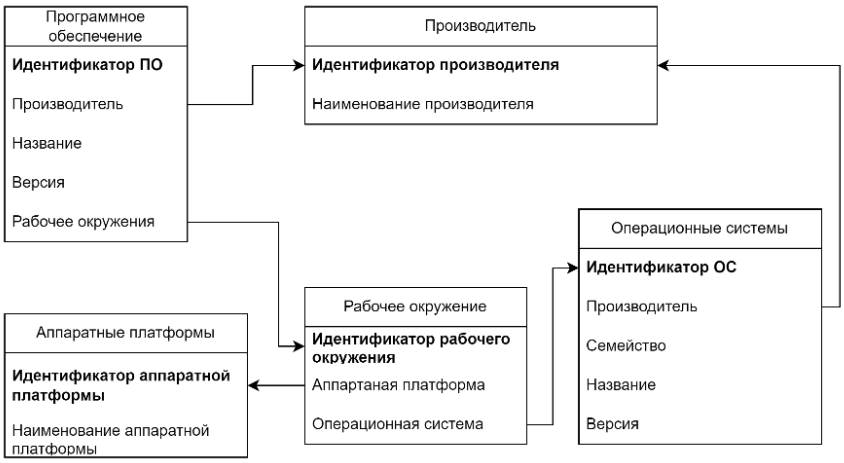

Рис. 1. Предположительная структура хранения данных СИИПО

Для наглядности стоит рассмотреть вариант обращения к СИИПО на примере, ранее рассмотренного PostgreSQL 16. Так информация, собранная RedCheck, в структурированном виде (JSON) может иметь следующий вид (рис. 2).

```
{
  "vendor": "PostgreSQL Global Development Group",
  "name" : "PostgreSQL 16",
  "version": "16.3-1",
  "OS": "Майкрософт Windows 11 Pro",
  "hardware": "64-bit"
}
```

Рис. 2. Структурированная информация о PostgreSQL 16

Структурированная информация о программном обеспечении используется для запроса к API СИИПО. На стороне регулятора происходит поиск по существующим записям СИИПО – комбинация значений атрибутов «Операционная система» и «Аппаратная платформа» формируют «Рабочее окружение», значение которого сокращает область поиска до совместимого ПО, а атрибуты программного обеспечения позволяют найти

однозначный идентификатор в сущности «Программное обеспечение». Таким образом, результатом выполнения запроса к API СИИПО является уникальный идентификатор программного обеспечения.

Идентификатор программного обеспечения можно представить в виде UUID (universally unique identifier), где каждое значение атрибута программного обеспечения будет закодировано в битовую последовательность фиксированной длины. Использование не самого значения атрибута, а сопоставленного ему стандартизированного кода позволит значительно сократить выделяемую для хранения память. Для некоторых атрибутов уже существуют стандартные значения, например, для операционной системы или аппаратной платформы. Для демонстрации идеи на рис. 3 приведен возможный способ генерации идентификатора программного обеспечения.

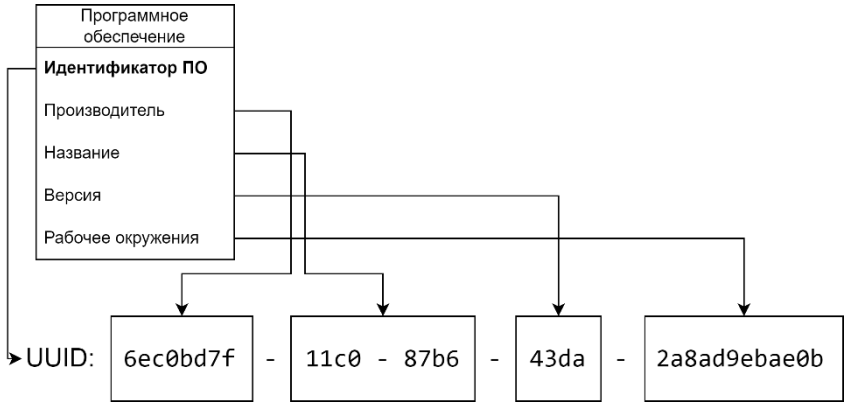

Рис. 3. Генерация UUID программного обеспечения

Технически возможно рассматривать два варианта получения идентификатора для последующей идентификации программного обеспечения:

1) отправка структурированной информации (рис. 2) о программном обеспечении через запрос к API СИИПО, генерация и поиск идентификатора на стороне СИИПО и возвращение полученного значения клиенту;

2) для снижения нагрузки на API СИИПО рекомендуется заранее утвердить явно стандартный алгоритм генерации идентификатора на основе структурированной информации о программном обеспечении. В таком случае можно получить идентификатор локально в информационной системе предприятия, без обращения к API СИИПО.

Таким образом, предполагается, что сбор данных о программном обеспечении и обогащение системы управления ИТ активами выполняет в информационной системе агент на конечном устройстве. Для гарантированного поддержания сведений, хранящихся в системе управления ИТ активами в актуальном состоянии, задача данного агента должна выполняться постоянно или периодически. Решение данной задачи является ответственностью специалистов на местах, так как структура хранения данных может изменяться в зависимости от применяемой системы управления ИТ активами.

Стоит отметить, что в рамках приведенной схемы предполагается что агент будет локально хранить в себе необходимую информацию для преобразования собранных данных о программном обеспечении в уникальные идентификаторы (ближайший аналог – CPE Dictionary). Для актуализации этих сведений предполагается периодический опрос СИИПО агентом. Такой подход необходим для снижения нагрузки на API СИИПО.

Полученный идентификатор будет применяться в приложении сканере уязвимостей (в качестве ближайшего аналога можно рассматривать ScanOVAL) для запроса к БДУ ФСТЭК [10] на уровне запроса к API (или ручного поиска) с целью однозначной идентификации уязвимого программного обеспечения (через уникальный идентификатор) и получение информации об актуальных для него уязвимостей – пассивный анализ ИТ активов на наличие уязвимостей в целевой ИТ инфраструктуре посредством отдельного приложения.

Данное приложение так же может выполняться периодически, тем самым пополняя систему управления ИТ активами данными об уязвимостях, которые присутствуют в ИТ инфраструктуре. Так же используя метод автоматизации из работы [21], можно обогатить записи об уязвимостях для их лучшей интеграции с требованиями методического документа «Методика оценки уровня критичности уязвимостей программных, программно-аппаратных средств» [22].

Обобщенный алгоритм применения предлагаемой авторами СИИПО в реальных процессах управления уязвимостями приведен на рис. 4.

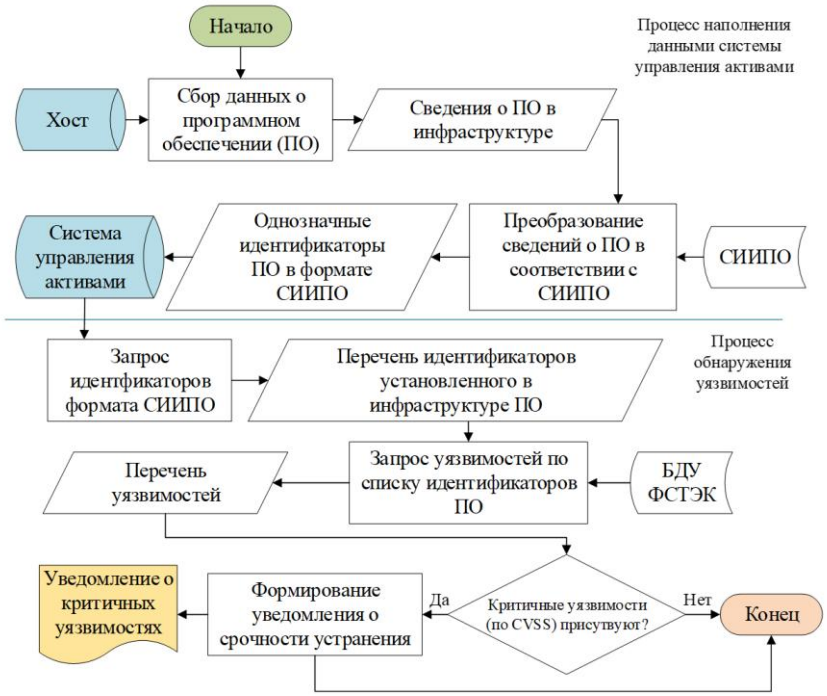

Рис. 4. Обобщенный алгоритм применения СИИПО в процессах управления уязвимостями

Исходя из такой логики применения СИИПО в реальных процессах управления уязвимостями можно выделить ряд преимуществ:

1) Соответствие показателям эффективности введенных регулятором. Так, например, при периодическом опросе БДУ ФСТЭК раз в час, в худшем случае срок получения сведений о критичных уязвимостях программного обеспечения имеющегося в целевой инфраструктуре будет составлять не более часа, что оставляет 23 часа на устранение уязвимости.

2) Уменьшение расходов на специализированные решения, такие как сканеры безопасности или решения класса Vulnerability Management, так как для функционирования требуется лишь система управления ИТ активами, в качестве которой может выступать решение с открытым исходным кодом, например, ранее упомянутый GLPI. Фактически, такой подход открывает возможность построения процесса управления уязвимостями для организации с низким уровнем зрелости ИБ или со значительно ограниченными ресурсами, например муниципальным учреждениям.

3) Потенциально возможно частичное решение проблемы с программным обеспечением собственной разработки [23], так как система управления ИТ активами может хранить сведения о зависимостях, применяемых в программном обеспечении собственной разработки, а в следствии и данные об уязвимостях в этих зависимостях.

Заключение. Проблема однозначного именования и идентификации программного обеспечения является значимой и актуальной, так как ее решение может значительно повлиять на процессы управления уязвимостями. Существующие решения, в частности CPE как наиболее успешный вариант, хоть и несли в себе идею их масштабного применения с течением времени пришли к тому, что они либо не применимы, либо могут лишь частично покрыть запросы специалистов. Помимо этого, некорректная эксплуатация систем привела к значительному усложнению их практического применяя. В таком случае идея создания национальной системы для государственного регулирования именования и идентификации программного обеспечения в перспективе открывает обширное количество возможностей для качественного развития управления уязвимостями в России и повышение защищенности ее информационных систем. В качестве направления развития данной работы можно выделить непосредственное формирование универсального идентификатора на основе данных, полученных на этапе сбора.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Методический документ ФСТЭК от 17 мая 2023 г. «Руководство по организации процесса управления уязвимостями в органе (организации)». – URL: https://fstec.ru/files/1096/---17--2023-/2011/---17--2023-.pdf (дата обращения: 10.05.2024).
2. Computer Security Incident Handling Guide. – URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf (дата обращения: 10.05.2024).
3. *Wang Z. et al.* Design and implementation of security vulnerability sharing platform based on web crawler // Proceedings of the 11th International Conference on Computer Engineering and Networks. – Springer Singapore, 2022. – P. 678-687.
4. *Bonomi S., Cuoci M., Lenti S.* A Version-Based Algorithm for Quality Enhancement of Automatically Generated Vulnerability Inventories // 2024 IEEE International Conference on Cyber Security and Resilience (CSR). – IEEE, 2024. – P. 76-81
5. *Luo J. et al.* CVECenter: Industry Practice of Automated Vulnerability Management for Linux Distribution Community // Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering. – 2024. – P. 329-339.
6. *Ghazo A.T.A.L., Kumar R*. ANDVI: Automated Network Device and Vulnerability Identification in SCADA/ICS by Passive Monitoring // IEEE Transactions on Systems, Man, and Cybernetics: Systems. – 2024.
7. *Leverett É., Rhode M., Wedgbury A.* Vulnerability Forecasting: Theory and practice // Digital Threats: Research and Practice. – 2022. – Vol. 3, No. 4. – P. 1-27
8. National Vulnerability Database. – URL: https://www.nist.gov/itl/nvd (дата обращения: 10.05.2024).
9. *Ncube Z.M.* Emerging Threats in Cybersecurity: Risk and Vulnerability Management // Journal of Innovative Technologies. – 2024. – Vol. 7, No. 1.
10. Реестр программного обеспечения. – URL: https://reestr.digital.gov.ru/reestr/ (дата обращения: 10.05.2024).
11. Банк данных угроз безопасности информации ФСТЭК. – URL: https://bdu.fstec.ru/threat (дата обращения: 10.05.2024).
12. *Селигеев С.В., Жуков В.Г.* О пассивном обнаружении уязвимостей программного обеспечения // Актуальные проблемы авиации и космонавтики: Сб. материалов X Международной научно-практической конференции, посвященной 100-летию академика М.Ф. Решетнева и Дню космонавтики. В 3-х т., Красноярск, 8–12 апреля 2023 года. – Красноярск: Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, 2024. – С. 277-279.
13. *Ecik H.* Comparison of active vulnerability scanning vs. passive vulnerability detection // 2021 International Conference on Information Security and Cryptology (ISCTURKEY). – IEEE, 2021. – P. 87-92.
14. OVALdb. – URL: https://www.altx-soft.ru/ovaldb/ (дата обращения: 10.05.2024).
15. *McClanahan K., Li Q.* Towards Automatically Matching Security Advisories to CPEs: String Similarity-based Vendor Matching // Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)-Workshop on Computing, Networking and Communications. – 2024.
16. *Hu W., Thing V.L.L.* CPE-Identifier: Automated CPE identification and CVE summaries annotation with Deep Learning and NLP // arXiv preprint arXiv:2405.13568. – 2024.
17. *Wåreus E., Hell M.* Automated CPE labeling of CVE summaries with machine learning // Detection of Intrusions and Malware, and Vulnerability Assessment: 17th International Conference, DIMVA 2020, Lisbon, Portugal, June 24–26, 2020, Proceedings 17. – Springer International Publishing, 2020. – P. 3-22.

18. *Михайлов В.А.* Российская система для полного цикла управления уязвимостями // Первая миля. – 2024. – № 3 (119). – С. 70-71. – DOI: 10.22184/2070-8963.2024.119.3.70.71. – EDN CIQOTG.
19. *Дорофеев А.В., Марков А.С.* Применение отечественных технологий для мониторинга информационной безопасности в условиях импортозамещения // Защита информации. Инсайд. – 2023. – № 3 (111). – С. 20-26. – EDN FDPTDW.
20. *Марунько А.С., Попов И.О*. Современная проблематика управления ИТ-активами в Российской Федерации // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2023. – № 4. – С. 91-95. – DOI: 10.37882/2223-2966.2023.04.24. – EDN AMAULO.
21. *Жуков В.Г., Селигеев С.В.* Автоматизация оценки уязвимостей программных, программно-аппаратных средств в целевой информационной системе // Прикаспийский журнал: управление и высокие технологии. – 2023. – № 4 (64). – С. 16-25. – DOI: 10.54398/20741707_2023_4_16. – EDN SRMLBD.
22. Методический документ ФСТЭК «Методика оценки уровня критичности уязвимостей программных, программно-аппаратных средств». – URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-28-oktyabrya-2022-g-2 (дата обращения: 10.05.2024).
23. *Селигеев С.В., Жуков В.Г.* О проблеме управления уязвимостями в программном обеспечении собственной разработки // Решетневские чтения: Матер. XXVII Международной научно-практической конференции, посвященной памяти генерального конструктора ракетно-космических систем академика М.Ф. Решетнева , Красноярск, 08–10 ноября 2023 года. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева", 2023. – С. 401-403. – EDN SGTSJB.

REFERENCES

1. Metodicheskiy dokument FSTEK ot 17 maya 2023 g. «Rukovodstvo po organizatsii protsessa upravleniya uyazvimostyami v organe (organizatsii)» [FSTEC Methodological Document of May 17, 2023 “Guidelines for Organizing the Vulnerability Management Process in a Body (Organization)”]. Available at: URL: https://fstec.ru/files/1096/---17--2023-/2011/---17--2023-.pdf (accessed 10 May 2024).
2. Computer Security Incident Handling Guide. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf (accessed 10 May 2024).
3. *Wang Z. et al.* Design and implementation of security vulnerability sharing platform based on web crawler, *Proceedings of the 11th International Conference on Computer Engineering and Networks*. Springer Singapore, 2022, pp. 678-687.
4. *Bonomi S., Cuoci M., Lenti S.* A Version-Based Algorithm for Quality Enhancement of Automatically Generated Vulnerability Inventories, *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2024, pp. 76-81
5. *Luo J. et al.* CVECenter: Industry Practice of Automated Vulnerability Management for Linux Distribution Community, *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering,* 2024, pp. 329-339.
6. *Ghazo A.T.A.L., Kumar R.* ANDVI: Automated Network Device and Vulnerability Identification in SCADA/ICS by Passive Monitoring, *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 2024.
7. *Leverett É., Rhode M., Wedgbury A.* Vulnerability Forecasting: Theory and practice, *Digital Threats: Research and Practice,* 2022, Vol. 3, No. 4, pp. 1-27
8. National Vulnerability Database. Available at: https://www.nist.gov/itl/nvd (accessed 10 May 2024).
9. *Ncube Z.M.* Emerging Threats in Cybersecurity: Risk and Vulnerability Management, *Journal of Innovative Technologies,* 2024, Vol. 7, No. 1.
10. Reestr programmnogo obespecheniya [Software registry]. Available at: https://reestr.digital.gov.ru/reestr/ (accessed 10 May 2024).
11. Bank dannykh ugroz bezopasnosti informatsii FSTEK [FSTEC information security threat database]. Available at: https://bdu.fstec.ru/threat (accessed 10 May 2024).
12. *Seligeev S.V., Zhukov V.G.* O passivnom obnaruzhenii uyazvimostey programmnogo obespecheniya [On passive detection of software vulnerabilities], *Aktual'nye problemy aviatsii i kosmonavtiki: Sb. materialov X Mezhdunarodnoy nauchno-prakticheskoy konferentsii, posvyashchennoy 100-letiyu akademika M.F. Reshetneva i Dnyu kosmonavtiki. V 3-kh t., Krasnoyarsk, 8–12 aprelya 2023 goda* [Actual problems of aviation and cosmonautics: Collection of materials of the X International scientific and practical conference dedicated to the 100th anniversary of academician M.F. Reshetnev and Cosmonautics Day. In 3 volumes, Krasnoyarsk, April 8–12, 2023]. Krasnoyarsk: Sibirskiy gosudarstvennyy universitet nauki i tekhnologiy imeni akademika M.F. Reshetneva, 2024, pp. 277-279.

13. *Ecik H.* Comparison of active vulnerability scanning vs. passive vulnerability detection, *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*. IEEE, 2021, pp. 87-92.
14. OVALdb. Available at: https://www.altx-soft.ru/ovaldb/ (accessed 10 May 2024).
15. *McClanahan K., Li Q.* Towards Automatically Matching Security Advisories to CPEs: String Similarity-based Vendor Matching, *Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)-Workshop on Computing, Networking and Communications,* 2024.
16. *Hu W., Thing V.L.L.* CPE-Identifier: Automated CPE identification and CVE summaries annotation with Deep Learning and NLP, *arXiv preprint arXiv:2405.13568,* 2024.
17. *Wåreus E., Hell M.* Automated CPE labeling of CVE summaries with machine learning, *Detection of Intrusions and Malware, and Vulnerability Assessment: 17th International Conference, DIMVA 2020, Lisbon, Portugal, June 24–26, 2020, Proceedings 17*. Springer International Publishing, 2020, pp. 3-22.
18. *Mikhaylov V.A.* Rossiyskaya sistema dlya polnogo tsikla upravleniya uyazvimostyami [Russian system for a full cycle of vulnerability management], *Pervaya milya* [Pervaya mile], 2024, No. 3 (119), pp. 70-71. DOI: 10.22184/2070-8963.2024.119.3.70.71. EDN CIQOTG.
19. *Dorofeev A.V., Markov A.S.* Primenenie otechestvennykh tekhnologiy dlya monitoringa informatsionnoy bezopasnosti v usloviyakh importozameshcheniya [Application of domestic technologies for monitoring information security in the context of import substitution], *Zashchita informatsii. Insayd* [Information protection], 2023, No. 3 (111), pp. 20-26. EDN FDPTDW.
20. *Marun'ko A.S., Popov I.O.* Sovremennaya problematika upravleniya IT-aktivami v Rossiyskoy Federatsii [Modern problems of IT asset management in the Russian Federation], *Sovremennaya nauka: aktual'nye problemy teorii i praktiki. Seriya: Estestvennye i tekhnicheskie nauki* [Modern science: current problems of theory and practice. Series: Natural and technical sciences], 2023, No. 4, pp. 91-95. DOI: 10.37882/2223-2966.2023.04.24. EDN AMAULO.
21. *Zhukov V.G., Seligeev S.V.* Avtomatizatsiya otsenki uyazvimostey programmnykh, programmno-apparatnykh sredstv v tselevoy informatsionnoy sisteme [Automation of vulnerability assessment of software, software and hardware in the target information system], *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Management and High Technologies], 2023, No. 4 (64), pp. 16-25. DOI: 10.54398/20741707_2023_4_16. EDN SRMLBD.
22. Metodicheskiy dokument FSTEK «Metodika otsenki urovnya kritichnosti uyazvimostey programmnykh, programmno-apparatnykh sredstv» [FSTEC Methodological Document "Methodology for Assessing the Criticality Level of Vulnerabilities in Software, Software and Hardware"]. Available at: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-28-oktyabrya-2022-g-2 (accessed 10 May 2024).
23. *Seligeev S.V., Zhukov V.G.* O probleme upravleniya uyazvimostyami v programmnom obespechenii sobstvennoy razrabotki [On the Problem of Vulnerability Management in In-House Software], *Reshetnevskie chteniya: Mater. XXVII Mezhdunarodnoy nauchno-prakticheskoy konferentsii, posvyashchennoy pamyati general'nogo konstruktora raketno-kosmicheskikh sistem akademika M.F. Reshetneva , Krasnoyarsk, 08–10 noyabrya 2023 goda* [Reshetnev Readings: Proceedings of the XXVII International Scientific and Practical Conference Dedicated to the Memory of the General Designer of Rocket and Space Systems, Academician M.F. Reshetnev, Krasnoyarsk, November 8-10, 2023]. Krasnoyarsk: Federal'noe gosudarstvennoe byudzhetnoe obrazovatel'noe uchrezhdenie vysshego obrazovaniya "Sibirskiy gosudarstvennyy universitet nauki i tekhnologiy imeni akademika M.F. Reshetneva", 2023, pp. 401-403. EDN SGTSJB.

Жуков Вадим Геннадьевич – Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева; e-mail: zhukov@sibsau.ru; г. Красноярск, Россия; тел.: +73912227639; к.т.н.; доцент.

Селигеев Сергей Викторович – e-mail: seligeevsergei@gmail.com; тел.: +79135845060; ассистент.

Zhukov Vadim Gennad`evich – Reshetnev Siberian State University of Science and Technology; e-mail: zhukov@sibsau.ru; Krasnoyarsk, Russia; phone: +73912227639; cand. of eng. sc.; associate professor.

Seligeev Sergej Viktorovich – e-mail: seligeevsergei@gmail.com; phone: +79135845060; assistant.