УДК 004.056.5: 004.822 Вестник УрФО № 2(40) / 2021, с. 31–43

Васильев В.И., Вульфин А.М., Кириллова А.Д., Никонов А.В.

DOI: 10.14529/secur210204

СИСТЕМА ОЦЕНКИ МЕТРИК ОПАСНОСТИ УЯЗВИМОСТЕЙ НА ОСНОВЕ ТЕХНОЛОГИЙ СЕМАНТИЧЕСКОГО АНАЛИЗА ДАННЫХ[1]

Предложена система прогнозирования компонент вектора метрик уязвимостей и количественной оценки степени опасности этих уязвимостей на основе анализа текстового описания с помощью инструментов обработки естественного языка (Natural Language Processing, NLP). Применение системы направлено на повышение оперативности оценки опасности выявляемых уязвимостей программно-аппаратного обеспечения автоматизированной системы управления технологическими процессами для соответствующего реагирования и принятия необходимых мер с целью обеспечения требуемого уровня кибербезопасности объектов и систем.

Ключевые слова: *информационная безопасность, угрозы, уязвимости, обработка естественного языка, CVSS, анализ текстов.*

Vasilyev V.I., Vulfin A.M., Kirillova A.D., Nikonov A.V.

SYSTEM FOR EVALUATING VULNERABILITY SEVERITY METRICS BASED ON SEMANTIC DATA ANALYSIS TECHNOLOGIES

A system for predicting the components of the vulnerability metrics vector and quantifying the severity of these vulnerabilities based on the analysis of the textual description using natural language processing (NLP) tools is proposed. The application of the system is aimed at increasing the efficiency of assessing the danger of the identified vulnerabilities of the software and hardware of the APCS for appropriate response and taking the necessary measures in order to ensure the required level of cybersecurity of objects and systems.

[1] Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-08-00668 и при финансовой поддержке Минобрнауки России (грант ИБ) № 1/2020.

Keywords: *information security, threats, vulnerabilities, natural language processing, CVSS, analysis of texts.*

Введение

Ежедневно регистрируются новые уязвимости программного и аппаратного обеспечения информационных систем (ИС), но их анализ и присвоение количественной оценки уровня опасности, по-прежнему, занимает продолжительное время (до трех месяцев). Согласно статистике компании Claroty [1], за 2020 год выявлено 893 уязвимостей, что на 24,72% больше, чем в 2019 году. Более 70% уязвимостей получили статус критических или высокую степень опасности.

В [2] на примере NVD (National Vulnerability Database – хранилище данных уязвимостей, основанное на стандартах правительства США) выполнен анализ зависимости времени появления эксплоита от компонент вектора метрик CVSS (Common Vulnerability Scoring System – общепринятый стандарт для определения степени опасности уязвимостей в программном обеспечении) уязвимостей. Показано, что существуют классы уязвимостей с очень коротким медианным временем появления эксплоитов (три дня). Отмечено, что временная задержка заполнения NIST (Национальный институт стандартов и технологий США) метрик CVSS после публикации уязвимости возросла с одного дня (для уязвимостей до 2017 года) до 19 дней (для уязвимостей, внесенных в базу в 2018 году). Напротив, среднее время появления эксплоита для уязвимостей сократилось с 296 дней в 2005 году до шести дней в 2018 году.

Важность наискорейшей оценки опасности уязвимости с момента ее регистрации в открытых источниках обусловлена высокой вероятностью ее эксплуатации злоумышленниками. Недостаток информации о степени опасности уязвимости и ее характеристиках осложняет планирование и проведение мероприятий по защите уязвимых ИС для специалистов. Следовательно, в ходе аудита ИС и инвентаризации программного обеспечения (ПО) важно иметь актуальную информацию по выявленным уязвимостям и количественным оценкам их опасности для эффективного планирования защитных мероприятий.

Целью работы является повышение эффективности (точности и оперативности) оценки опасности уязвимостей с помощью прогнозирования векторов метрики на основе анализа текстового описания.

1. Анализ применения технологий Text Mining в задаче оценки опасности уязвимостей

Исследованию возможностей методов семантического анализа текстов (Text Mining) для решения задач анализа и прогнозирования опасности уязвимостей ПО непосредственно по их текстовым описаниям, хранящимся в базах данных (БД) уязвимостей, посвящен ряд работ [3–12]. Так, в [3] авторы проанализировали появление новых записей CVE (база данных общеизвестных уязвимостей информационной безопасности) за 23 месяца и установили, что в среднем существует 132-дневный разрыв между объявлением уязвимости MITRE (некоммерческая организация, специализирующаяся в области системной инженерии) и моментом, когда NIST определяет уровень опасности уязвимости и метрики CVSS. Предложена система анализа уязвимостей, позволяющая прогнозировать эксплуатацию уязвимости и выполнять оценку компонент вектора метрик CVSS, используя текстовые данные обсуждения Twitter, собранные за три дня после даты первого упоминания уязвимости.

В [4] предложен новый подход к структурированию описаний уязвимостей, позволяющий в явном виде выделить условия реализации уязвимости как последовательность определенных событий – действий со стороны пользователей и атакующих, что важно в первую очередь для понимания характера уязвимости и способа ее устранения.

Авторы статьи [5] решают задачу оценки критичности (опасности) уязвимостей в два этапа, на первом из которых осуществляется векторизация (Word Embedding) текстовых описаний уязвимостей, а на втором производится классификация полученных векторов (наборов уникальных признаков описаний) с помощью методов машинного обучения.

В работе [6] используется аналогичный подход к оценке степени критичности уязвимостей, предполагающий на начальном этапе формирование словаря векторов слов для каждого описания уязвимости (для этого используется модель Skip-Gram алгоритма Word2Vec), а затем извлечение признаков текста на уровне предложений и классификация полученного вектора описаний уязвимости с помощью сверточной нейронной сети.

Статья [7] посвящена задаче ранжирования (приоритезации) уязвимостей ПО, для решения которой предложена следующая процедура: а) оформляется корпус CVE-описаний уязвимостей из базы данных NVD; б) производится конвертация CVE-описаний в «мешок слов» (модель CBOW); в) осуществляется ранжирование уязвимостей по степени важности с применением алгоритма TextRank, основанного на оценке семантической близости предложений в тексте.

В работе [8] использованы модели машинного обучения и обработки естественного языка для прогнозирования последствий кибератак. Представлена модель векторизации текстовых описаний новых кибератак и прогнозирования последствий для конечных пользователей. Создан набор данных о кибератаках с указанием их технических и нетехнических последствий. При помощи методов вложения слов (Doc2Vec) подготовлены данные для ансамбля моделей машинного обучения (LinearSVC, NB, MLP), оценка качества прогнозирования составила до 60 %.

В работе [9] предложена модель для прогнозирования компонент базового вектора CVSS с возможностью объяснения прогноза, которая использует текстовые описания новых уязвимостей. Применяются технологии Bag-of-Word и оценка энтропии вхождения слов в текстовые описания.

В работе [10] рассмотрен метод оценки CVSS на основе текстовых описаний уязвимостей из базы данных OSVDB. Выполнено извлечение и предобработка текстового описания уязвимостей, использованы методы LDA и PCA для уменьшения размеров вектора признаков, применены алгоритмы SVM и Random Forest, а также нечеткие системы для прогнозирования оценок компонент вектора CVSS. Лучший предиктор был получен с помощью нечеткой системы с точностью прогноза по шкале CVSS на уровне 88 %.

Анализ публикаций также показал, что прогнозирование оценки метрики CVSS опасностей уязвимостей выполнено для англоязычных описаний уязвимостей. Открытым остается вопрос о повышении оперативности оценки опасности уязвимостей для программных продуктов и систем, распространенных на локальных рынках, и для которых описания уязвимостей представлены, например, на русском языке. На примере Банка данных угроз безопасности информации (БДУ) ФСТЭК России рассмотрим возможность прогнозирования компонент вектора базовой метрики CVSS 2.0/3.0 и оценки уровня опасности уязвимостей на основе технологий Text Mining.

2. Архитектура системы оценки опасности уязвимостей на основе технологий интеллектуального анализа данных

Компоненты базовой метрики CVSS 2.0/3.0 представлены в таблице 1.

Оценка опасности уязвимости на основе базовой метрики (CVSS 2.0) определяется как:

$$BaseScore = (0{,}6 \cdot Impact + 0{,}4 \cdot Exploitability - 1{,}5) \cdot f(Impact), \quad (1)$$

где оценка воздействия:

$$Impact = 10{,}41 \cdot (1 - (1 - C) \cdot (1 - I) \cdot (1 - A)), \quad (2)$$

оценка возможности эксплуатации:

$$Exploitability = 20 \cdot AC \cdot Au \cdot AV, \quad (3)$$

$f(Impact) = 0$, если $Impact = 0$;

$f(Impact) = 1{,}176$ в других случаях.

Следовательно, возможно два подхода для оценки BaseScore CVSS 2.0/3.0:

– построение ансамбля предикторов для оценки отдельных значений компонент вектора (AV, AC, Au, C, I, A) по формализованному текстовому описанию с последующим расчетом по (1)–(3) результирующего значения;

– построение модели регрессии для непосредственной оценки результирующего значения по формализованному текстовому описанию.

Архитектура предлагаемой системы для прогнозирования оценки опасности уязвимости CVSS 2.0/3.0 представлена на рис. 1.

Первый этап работы системы связан со сбором и агрегацией специализированных новостных рассылок и тематических ресурсов в виде слабоструктурированных текстовых данных для построения документоориентированной БД$_1$ (MongoDB) с привязкой записей к ключу CVE-ID из БДУ ФСТЭК и NVD (CVE, CWE (БД недостатков ПО, которые могут быть использованы злоумышленниками), CPE (формальный язык описания всех возможных продуктов, операционных систем и аппаратных устройств при описании уязвимостей)). Процесс сбора данных регулируется специалистами по информационной безопасности с применением Threat Intelligence.

Затем собранные текстовые данные подвергаются предобработке и нормализации: символьная фильтрация, токенизация, фильтрация на основе стоп-словарей, лемматизация – на основе технологии конвейеризации NLP-Pipe.

Далее строится ансамбль Text Mining мо-

Таблица 1

Компоненты базовой метрики CVSS 2.0/3.0

Метрика	Буквенное обозначение	Возможные значения
Способ получения доступа (Access Vector – v2) (Attack Vector – v3)	AV	Physical (v3.0)
		Local (v2.0, 3.0)
		Adjacent (v2.0, 3.0)
		Network (v2.0, 3.0)
Сложность эксплуатации уязвимости (Access Complexity – v2) (Attack Complexity – v3)	AC	Lower (v2.0, 3.0)
		High (v2.0, 3.0)
		Medium (v2.0)
Показатель аутентификации (Authentication – v2) (Privileges Required – v3)	Au (v2.0) Pr(v3.0)	None
		Low (v3.0)
		High (v3.0)
		Single (v2.0)
		Multiple (v2.0)
Влияние на конфиденциальность	C	None
		Low (v3.0)
		High (v3.0)
		Partial (v2.0)
		Complete (v2.0)
Влияние на целостность	I	None
		Low (v3.0)
		High (v3.0)
		Partial (v2.0)
		Complete (v2.0)
Влияние на доступность	A	None
		Low (v3.0)
		High (v3.0)
		Partial (v2.0)
		Complete (v2.0)
Необходимость взаимодействия с пользователем (User Interaction)	UI	None (v3.0)
		Required (v3.0)
Границы эксплуатации (Scope)	S	Unchanged (v3.0)
		Changed (v3.0)

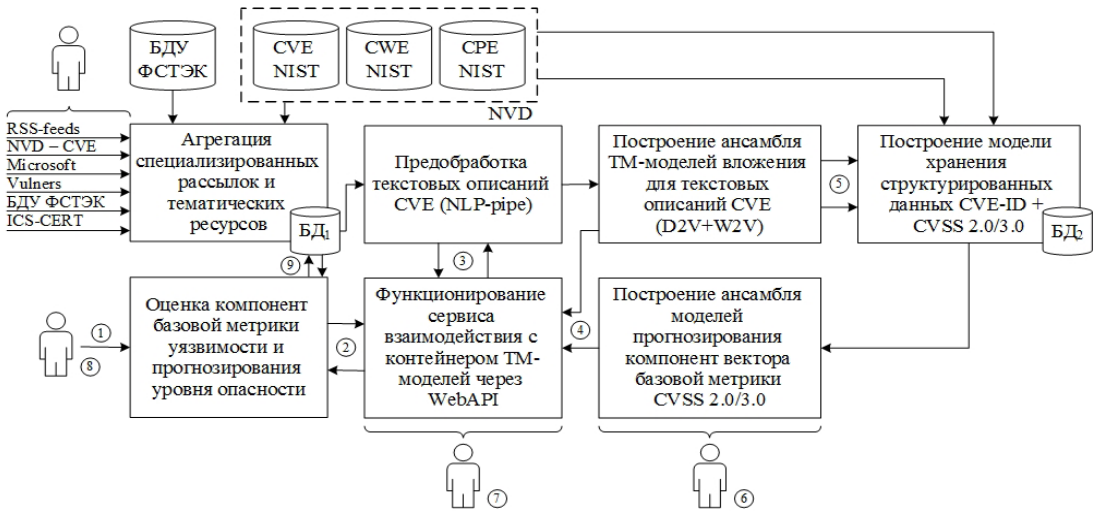

Рис. 1. Архитектура системы оценки опасности уязвимостей на основе технологий интеллектуального анализа данных

делей вложения на основе очищенных текстовых описаний уязвимостей с привязкой к ключу CVE-ID как взвешенная комбинация Doc2Vec и Word2Vec моделей. С помощью D2V и W2V моделей (5) формируется вектор вложений текстовых описаний для БД2, предназначенной для хранения структурированных данных {CVE-ID, вектор CVSS 2.0/3.0, Text Embedded Vector}.

На следующем шаге конструируется ансамбль моделей для прогнозирования компонент вектора базовой метрики CVSS 2.0/3.0.

Подготовленные модели прогнозирования вектора CVSS 2.0/3.0 помещаются в контейнер (4) для размещения на сервере для обработки запросов от пользователей системы через WebAPI.

Второй этап предполагает обработку запросов (1) по оценке уровня опасностей выявленных уязвимостей, для которой отсутствует оценка CVSS и размеченный вектор базовой метрики от специалиста по информационной безопасности (8), проводящего аудит ИС. Выполняется передача (9 и 2) текстовых описаний уязвимости и/или CVE-ID в одну из баз уязвимостей, а также проводится подготовка (3) его нормализованного и предобработанного текстового описания.

Поддержка адекватного состояния и дообучение моделей прогнозирования осуществляется инженером по работе с моделями машинного обучения (machine learning, ML) (6). Функционирование контейнера ML-моделей на сервере и обработка WebAPI запросов обеспечивается инженером поддержки Web-сервиса (7).

Таблица 2

Структура конвейера NLP-Pipe

Этап	Шаги	Действия	Инструменты
Предобработка	символьная фильтрация	Удаление нерелевантных символов, разворачивание сокращений, очистка от html-тегов	Набор из 40 регулярных выражений и библиотека BuitifulSoup
	токенизация	Разбивка текста на токены с помощью предобученной для русского языка нейросетевой модели	Razdel [13] (фреймворк Natasha)
	фильтрация нерелевантных токенов	Удаление дат, цифр, чисел, ссылок, сокращений	Регулярные выражения
Нормализация	лемматизация	Приведение слов в исходную форму с помощью предобученной нейросетевой модели	Morph (фреймворк Natasha)
Постобработка	частеречная фильтрация	Остаются только существительные, глаголы, прилагательные, наречия, местоимения	Morph (фреймворк Natasha)
	фильтрация на основе стоп-словарей	Фильтрация нерелевантных лемм с помощью составного стоп-словаря, включающего наиболее часто встречающиеся слова корпуса текстов	NLTK-russian, NLTK-english
	Формирование документа-строки	Объединение лемм в нормализованную строку-документ	

3. Анализ корпуса русскоязычных текстов – описаний уязвимостей БДУ ФСТЭК

Корпус текстов для анализа построен из 31384 агрегированных текстовых описаний уязвимостей на русском языке из БДУ ФСТЭК России. Составное текстовое поле включает данные о характеристике уязвимости и вариантах ее эксплуатации. Структура конвейера NLP-Pipe для обработки данных представлена в таблице 2.

Предварительный анализ корпуса текстов позволит оценить структуру корпуса и возможности применения моделей для построения предикторов. С помощью библиотеки Gensim выполним частотный анализ и оценку длины отдельных текстовых документов (рис. 2).

Далее выполним тематическое моделиро-

Рис. 2. Распределение количества слов в документах

вание на основе алгоритма латентного размещения Дирихле (LDA), которое позволит оценить возможность представления текстовых документов в виде смеси автоматически выделенных тем, и привязкой каждого слова к одной из них. Тематическая модель позволяет оценить структурированность текстов и потенциал их группировки по степени семантической близости. Использован вариант LDA на основе иерархической байесовской модели: на первом уровне – компоненты модели соответствуют «темам»; на втором уровне – мультиномиальной переменной с априорным распределением Дирихле, которые задают «распределение тем» в документе. Используются значения матрицы близости, основанной на частотных характеристиках документов и лексических единиц для первых четырех выделенных тем, что позволяет оценить словарный состав и частотное распределение слов для каждой из них (рис. 3).

Сложность (обобщающую способность) модели LDA составляет 5,926, а ее когерентность (мера интерпретируемости – оценка согласованности темы) 0,459, что позволяет

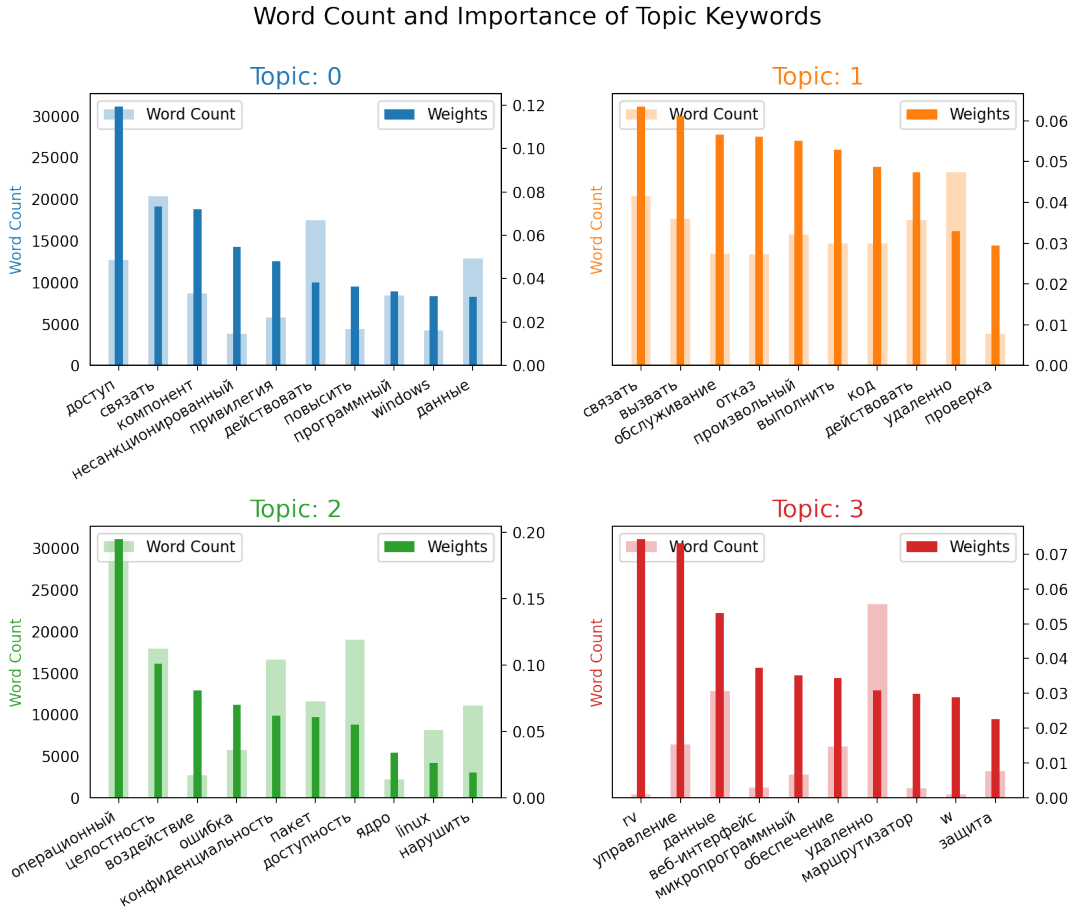

Рис. 3. Частотные характеристики первых четырех выделенных тем

сделать вывод о наличии структуры у корпуса текстов и возможности построения предикторов, реализующих задачу классификации по компонентам вектора метрики CVSS 2.0/3.0. С помощью t-distributed stochastic neighbor embedding, (стохастическое вложение соседей с распределением Стьюдента t-SNE) выполним понижение размерности

признакового пространства и визуализацию в пространстве двух переменных распределения документов по первым 4 темам (рис. 4). Компактные группы объектов хорошо отделимы друг от друга, однако, наличие характерных выбросов у каждой группы требует дальнейшего анализа и изменения алгоритма постфильтрации на основе расширяемого стоп-словаря, но не препятствует построению предикторов.

Далее для формализации признаков строится нейросетевая модель векторного вложения для текстовых документов Distributed memory (PV-DM) Doc2Vec [14] (таблица 3).

Для каждого документа корпуса с помощью обученной D2V модели строится вектор

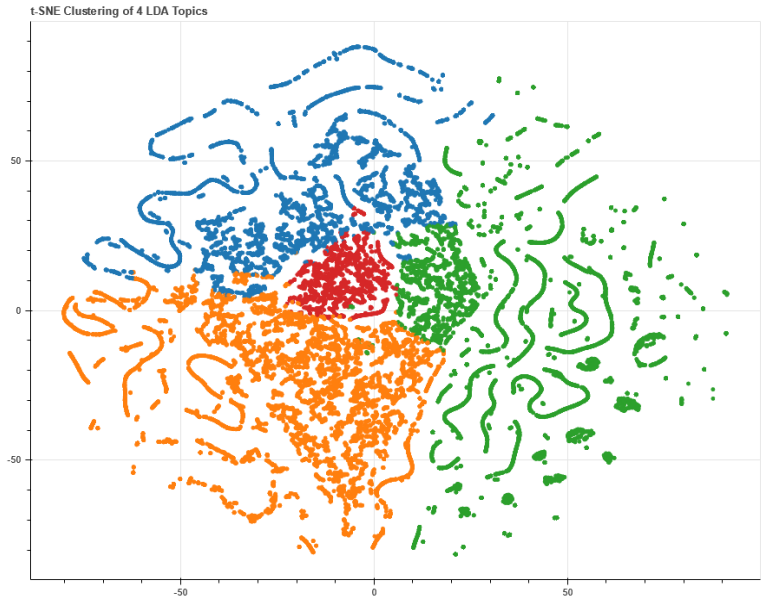

Рис. 4. t-SNE визуализация распределения для четырех основных тем

Таблица 3

Параметры Distributed memory (PV-DM) Doc2Vec Model

Параметр	Значение
Размерность вектора признаков	100
Размер окна анализа	5
Минимальная частота встречаемости слова для включения в модель	2
Количество эпох обучения	100

формальных признаков, на основе которого может быть оценена семантическая близость документов как косинус мера расстояния между векторами [15]. Это позволяет выполнять более качественный по сравнению с частотно-словарным (TF-IDF) поиск наиболее близких по смыслу документов. Предварительно рассчитанная разреженная матрица попарных расстояний позволяет существенно ускорить процедуру поиска и группировки уязвимостей. Пример поиска семантически близких описаний уязвимостей представлен в таблице 4.

4. Экспериментальная оценка опасности уязвимостей на основе технологий интеллектуального анализа данных

Для оценки BaseScore CVSS 2.0 рассмотрим два сценария:

– построение ансамбля предикторов для оценки отдельных значений компонент вектора (AV, AC, Au, C, I, A) по формализованному текстовому описанию с последующим расчетом оценки уровня опасности (таблица 5);

– построение модели регрессии для непосредственной оценки результирующего значения по формализованному текстовому описанию.

Построение ансамбля предикторов выполнено для 75% доступных документов с формальным вектором признаков, сформированным с помощью D2V модели. Оптимизация гиперпараметров используемых моделей предикторов выполнена с помощью процедуры перебора по сетке (GridSearch) с применением перекрестной проверки с разбие-

Таблица 4

Семантически близкие текстовые описания уязвимостей в порядке убывания косинус-меры

№	Документ	Мера близости
0	Уязвимость микропрограммного обеспечения программируемого логического контроллера Schneider Electric Modicon Quantum, позволяющая злоумышленнику получить авторизованный доступ к устройству. Микропрограммное обеспечение модуля 140NOE77111 контроллера Schneider Electric Modicon Quantum содержит множество пар логин: пароль, предустановленных по умолчанию. Это позволяет любому пользователю, имеющему доступ к устройству по протоколу FTP, получить авторизованный доступ к устройству	1
1	Уязвимость FTP-сервера микропрограммного обеспечения программируемых логических контроллеров Schneider Electric Modicon Premium, Modicon Quantum, Modicon M340 и Modicon BMXNOR0200, позволяющая нарушителю получить доступ к устройству. Уязвимость FTP-сервера микропрограммного обеспечения программируемых логических контроллеров Schneider Electric Modicon Premium, Modicon Quantum, Modicon M340 и Modicon BMXNOR0200 связана с использованием предустановленных учетных данных. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, получить доступ к устройству	0,6742
2	Уязвимость микропрограммного обеспечения устройства SIEMENS LOGO!8, связанная с неправильным контролем доступа, позволяющая нарушителю получить доступ к устройству. Уязвимость микропрограммного обеспечения программируемого логического контроллера SIEMENS LOGO!8 связана с неправильным контролем доступа. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, использовать перехваченный идентификатор сессии для доступа к устройству (даже после окончания сессии легитимным пользователем)	0,6724
…	…	…
25	Уязвимость микропрограммного обеспечения программируемого логического контроллера Modicon, связанная с раскрытием информации, позволяющая нарушителю получить доступ к конфиденциальной информации. Уязвимость микропрограммного обеспечения программируемого логического контроллера Modicon связана с раскрытием информации. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, получить доступ к конфиденциальной информации протокола SNMP при чтении блоков памяти контроллера с использованием протокола Modbus	0,5761

нием на 5 блоков. Параметры подбора приведены в таблице 5.

Результаты работы каждого предиктора на обучающей и тестовой выборках (25% исходных документов) для всех компонент вектора базовой метрики приведены в таблице 6 и на рис. 5.

Оценка классификаторов выполнена с помощью следующих метрик:

– Accuracy (точность) показывает долю правильных классификаций.

– Precision (точность) показывает долю объектов класса среди всех объектов, выделенных классификатором

– Recall (полнота) отражает долю найденных объектов класса от общего числа объектов класса.

– F_1 – среднее гармоническое Precision и Recall.

Второй сценарий подразумевает построение модели регрессии на основе ансамбля решающих деревьев (Random Forest) с оптимизацией гиперпараметров с помощью процедуры перебора по сетке (GridSearch) с применением перекрестной проверки с разбиением на 5 блоков. Результирующая модель ансамбля включает 500 решающих деревьев с максимальной глубиной 8.

Для обучающей выборки среднеквадратичная ошибка (Root Mean Square Error) составила 0,669, а для тестовой – 1,316.

Прогноз оценки уровня опасности уязвимости для 50 примеров из обучающей выборки и 50 примеров тестовой выборки представлен на рис. 6, где маркер «круг» – исходное значение, маркер «крест» – предсказанного, ось ординат – уровень опасности уязвимости.

5. Анализ результатов

Анализ таблицы 4 показывает, что выделенные уязвимости семантически близки к исходному документу, предварительно построенная матрица попарных расстояний близости описаний позволяет за константное время выполнять поиск близких описаний уязвимостей и их ранжирование для представления специалисту, проводящему аудит защищенности системы.

Таблица 5

Параметры моделей предикторов

Классификатор	Пространство гиперпараметров		Параметры обучения		
			Моделей	K-fold cross-validation	Среднее время, m
SGD (SVM) Классификатор на основе машины опорных векторов	tol (точность)	1e-3	576	5	35
	loss (функция потерь)	hinge, modified_huber			
	max_iter (максимальное количество и итераций)	10, 100, 1000			
	alpha (коэффициент регуляризации)	1e-6, 1e-4, 1e-2, 1e-1			
	learning_rate (алгоритм изменения градиентного шага)	optimal, invscaling			
	eta0 (начальный градиентный шаг)	1e-5, 1e-3, 1e-1, 1			
	penalty (метод регуляризации)	l1, l2, elasticnet			
SGD (LR) Классификатор на основе модели линейной регрессии	tol	1e-3	288	5	19,3
	loss	hinge, modified_huber			
	max_iter	10, 100, 1000			
	alpha	1e-6, 1e-4, 1e-2, 1e-1			
	learning_rate	optimal, invscaling			
	eta0	1e-5, 1e-3, 1e-1, 1			
	penalty	l1, l2, elasticnet			
KNeighbors Классификатор k ближайших соседей	n_neighbors (количество соседей для использования по умолчанию для запросов k соседей)	5	12	5	3,9
	weights (весовая функция, используемая в прогнозировании)	distance			
RandomForest Классификатор на основе комитета случайных деревьев решений	max_depth (максимальная глубина дерева. Если None, то узлы расширяются до тех пор, пока все листья не станут чистыми или пока все листья не будут содержать менее min_samples_split выборок)	None	60	5	26,3
	min_samples_split (минимальное количество выборок, необходимое в узле, чтобы вызвать расщепление узла)	2			
	n_estimators (количество деревьев в ансамбле)	500			

Таблица 6

Результаты работы предикторов на обучающей и тестовой выборках

Компонент	Классификатор	Обучающая выборка				Тестовая выборка			
		F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall
AV	SGD (SVM)	0,948	0,958	0,939	0,958	0,948	0,958	0,939	0,958
	SGD (LR)	0,744	0,804	0,818	0,804	0,744	0,804	0,816	0,804
	KNeighbors	1,000	1,000	1,000	1,000	0,963	0,965	0,964	0,965
	RandomForest	1,000	1,000	1,000	1,000	0,948	0,953	0,955	0,953
AC	SGD (SVM)	0,588	0,645	0,590	0,645	0,580	0,640	0,581	0,640
	SGD (LR)	0,555	0,652	0,594	0,652	0,552	0,650	0,584	0,650
	KNeighbors	1,000	1,000	1,000	1,000	0,759	0,766	0,759	0,766
	RandomForest	1,000	1,000	1,000	1,000	0,737	0,764	0,778	0,764
Au	SGD (SVM)	0,786	0,838	0,779	0,838	0,782	0,836	0,770	0,836
	SGD (LR)	0,771	0,843	0,710	0,843	0,771	0,843	0,710	0,843
	KNeighbors	1,000	1,000	1,000	1,000	0,898	0,901	0,896	0,901
	RandomForest	0,988	0,989	0,989	0,989	0,872	0,889	0,881	0,889
C	SGD (SVM)	0,661	0,673	0,665	0,673	0,652	0,665	0,657	0,665
	SGD (LR)	0,506	0,581	0,606	0,581	0,495	0,571	0,582	0,571
	KNeighbors	1,000	1,000	1,000	1,000	0,770	0,773	0,771	0,773
	RandomForest	1,000	1,000	1,000	1,000	0,772	0,779	0,781	0,779
I	SGD (SVM)	0,700	0,707	0,698	0,707	0,694	0,701	0,692	0,701
	SGD (LR)	0,579	0,637	0,677	0,637	0,569	0,630	0,671	0,630
	KNeighbors	1,000	1,000	1,000	1,000	0,778	0,783	0,778	0,783
	RandomForest	1,000	1,000	1,000	1,000	0,773	0,782	0,779	0,782
A	SGD (SVM)	0,583	0,660	0,617	0,660	0,581	0,658	0,612	0,658
	SGD (LR)	0,508	0,608	0,562	0,608	0,516	0,615	0,583	0,615
	KNeighbors	1,000	1,000	1,000	1,000	0,780	0,787	0,780	0,787
	RandomForest	1,000	1,000	1,000	1,000	0,770	0,785	0,783	0,785

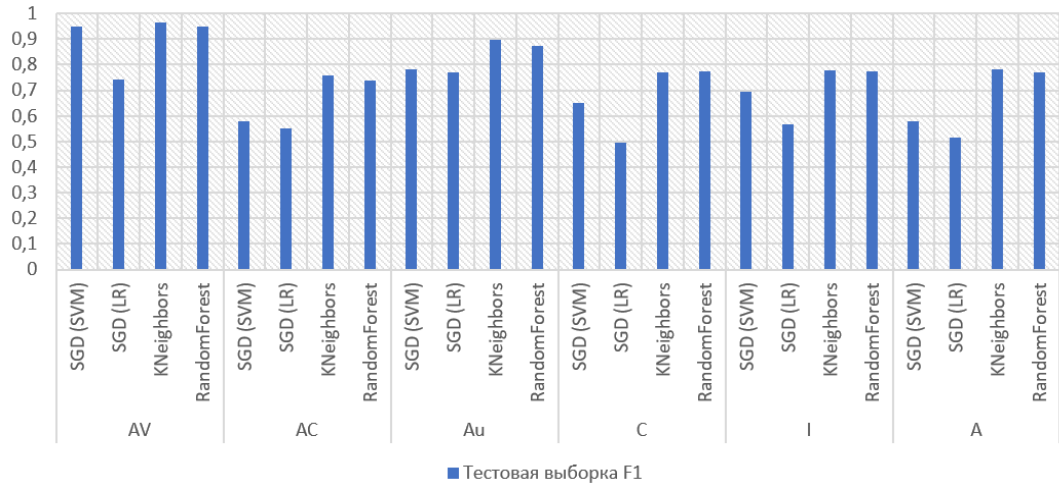

Рис. 5. Оценка F1 меры для тестовой выборки по каждому классификатору и компоненту метрики

Анализ таблицы 6 показывает, что ансамбль предикторов позволяет получить оценку компонент вектора базовой метрики новых уязвимостей на уровне значения меры F_1 = 0,70-0,75 для тестовой выборки, что свидетельствует о хорошей обобщающей способности предлагаемого решения.

Модель регрессии на основе ансамбля решающих деревьев позволяет непосредственно оценивать уровень опасности уязвимости, но без определения компонент базовой метрики.

Заключение

Предлагаемый подход основан на применении технологий интеллектуального анализа описаний уязвимостей на естественном

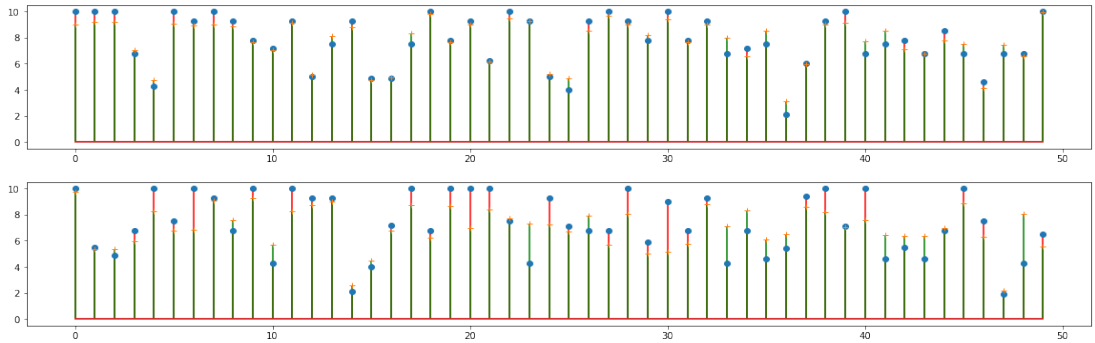

Рис. 6. Прогноз оценки уровня опасности уязвимости для 50 примеров из обучающей выборки (вверху) и 50 примеров тестовой выборки (внизу).

языке. Отличительной особенностью является использование построение модели вложения слов и описаний уязвимостей и композиции классификаторов, выполняющих оценку компонент вектора метрики уязвимости согласно стандарту CVSS. Применение предлагаемого подхода позволит получить оценку метрики опасности (и ее компонент) зарегистрированной уязвимости на основе анализа семантической близости текстового описания к уже имеющимся в реестре записям.

Практическая значимость обусловлена повышением эффективности (точности и оперативности) оценки метрик опасности уязвимостей с возможностью интеграции в систему аудита и инвентаризации для оперативного принятия мер по защите от новых уязвимостей.

Литература

1. Claroty Biannual ICS Risk & Vulnerability Report: 2H 2020. [Электронный ресурс] URL: https://security.claroty.com/biannual-ics-risk-vulnerability-report-2H-2020 (дата обращения: 10.04.2021)

2. Feutrill A. et al. The effect of common vulnerability scoring system metrics on vulnerability exploit delay // 2018 Sixth International Symposium on Computing and Networking (CANDAR). IEEE, 2018. P. 1–10.

3. Chen H. et al. VEST: A System for Vulnerability Exploit Scoring & Timing // IJCAI, 2019. P. 6503–6505.

4. Urbanska M., Ray I., Home A.E., Roberts M. Structuring a Vulnerability Description for Comprehensive Single System Security Analysis // RMCWiC'2012. [Электронный ресурс]. URL: www.cs.colostate.edu/psysec/papers/urbanskaRMCWiC`2012.pdf (дата обращения: 10.04.2021)

5. Spanos G., Angeis L., Toloudis D. Assessment of Vulnerability Severity using Text Mining // Proceedings of the 21st Pan-Hellenic Conference, Sept.2017, Larissa, Greece. P. 1–6.

6. Han Z., Li X., Xing Z., LiuH., Feng Z. Learning to Predict Severity of Software Vulnerability Description // Proceedings of the 2017 International Conference on Software Maintenance and Evolution (ICSME), Shanghai, China, Nov.2017. P. 125–136.

7. Lee Y., Shin S. Toward Semantic Assessment of Vulnerability Severity: A Text Mining Approach // Proceedings of ACM CIKM Workshop (EYRE' 18), 2018. [Электронный ресурс]. URL: https://www.CEUR-WS.org/Vol1-2482/papers.pdf (дата обращения: 10.04.2021)

8. Datta P. et al. Cyber-Attack Consequence Prediction //arXiv preprint arXiv:2012.00648. 2020.

9. Elbaz C., Rilling L., Morin C. Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure // Proceedings of the 15th International Conference on Availability, Reliability and Security, 2020. P. 1–10.

10. Khazaei A., Ghasemzadeh M., Derhami V. An automatic method for CVSS score prediction using vulnerabilities description // Journal of Intelligent & Fuzzy Systems, 2016. Vol. 30, No. 1. P. 89–96.

11. Лаврентьев А.М. и др. Сравнительный анализ специальных корусов текстов для задач безопасности // Вопросы кибербезопасности. 2020. № 3(37). С. 54–60.

12. Селифанов В.В., Юракова Я.В., Картамов И.Н. Методика автоматизированного выявления взаимосвязей уязвимостей и угроз безопасности информации в информационных системах // Интерэкспо Гео-Сибирь. 2018. С. 271–276.

13. Rule-based token, sentence segmentation for Russian language [Электронный ресурс] URL: https://github.com/natasha/razdel (дата обращения: 10.04.2021)

14. Mendsaikhan O. et al. Identification of cybersecurity specific content using the Doc2Vec language

model // 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). IEEE, 2019. Vol. 1. P. 396–401.

15. Бондарчук Д.В. Векторная модель представления знаний на основе семантической близости термов // Вестник ЮрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 7. С. 73–83.

References

1. Claroty Biannual ICS Risk & Vulnerability Report: 2H 2020. Available at: https://security.claroty.com/biannual-ics-risk-vulnerability-report-2H-2020 (accessed April 10, 2021).

2. Feutrill A. et al. The effect of common vulnerability scoring system metrics on vulnerability exploit delay // 2018 Sixth International Symposium on Computing and Networking (CANDAR). IEEE, 2018. pp. 1-10.

3. Chen H. et al. VEST: A System for Vulnerability Exploit Scoring & Timing // IJCAI, 2019. pp. 6503-6505.

4. Urbanska M., Ray I., Home A.E., Roberts M. Structuring a Vulnerability Description for Comprehensive Single System Security Analysis // RMCWiC'2012. Available at: www.cs.colostate.edu/psysec/papers/urbanskaRMCWiC`2012.pdf (accessed April 10, 2021).

5. Spanos G., Angeis L., Toloudis D. Assessment of Vulnerability Severity using Text Mining // Proceedings of the 21st Pan-Hellenic Conference, Sept.2017, Larissa, Greece. pp. 1-6.

6. Han Z., Li X., Xing Z., LiuH., Feng Z. Learning to Predict Severity of Software Vulnerability Description // Proceedings of the 2017 International Conference on Software Maintenance and Evolution (ICSME), Shanghai, China, Nov.2017. pp. 125-136.

7. Lee Y., Shin S. Toward Semantic Assessment of Vulnerability Severity: A Text Mining Approach // Proceedings of ACM CIKM Workshop (EYRE ' 18), 2018. Available at: https://www.CEUR-WS.org/Vol1-2482/papers.pdf (accessed April 10, 2021).

8. Datta P. et al. Cyber-Attack Consequence Prediction //arXiv preprint arXiv:2012.00648. 2020.

9. Elbaz C., Rilling L., Morin C. Fighting N-day vulnerabilities with automated CVSS vector prediction at disclosure // Proceedings of the 15th International Conference on Availability, Reliability and Security, 2020. pp. 1-10.

10. Khazaei A., Ghasemzadeh M., Derhami V. An automatic method for CVSS score prediction using vulnerabilities description // Journal of Intelligent & Fuzzy Systems, 2016. Vol. 30, no. 1. pp. 89-96.

11. Lavrent'ev A.M. i dr. Sravnitel'nyj analiz special'nyh korusov tekstov dlya zadach bezopasnosti // Voprosy kiberbezopasnosti. 2020. № 3(37). S. 54-60.

12. Selifanov V.V., YUrakova YA.V., Kartamov I.N. Metodika avtomatizirovannogo vyyavleniya vzaimosvyazej uyazvimostej i ugroz bezopasnosti informacii v informacionnyh sistemah // Interekspo Geo-Sibir'. 2018. S. 271-276.

13. Rule-based token, sentence segmentation for Russian language Available at: URL: https://github.com/natasha/razdel (accessed April 10, 2021).

14. Mendsaikhan O. et al. Identification of cybersecurity specific content using the Doc2Vec language model // 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). IEEE, 2019. Vol. 1. pp. 396-401.

15. Bondarchuk D.V. Vectornaya model' predstavleniya znanij na osnove semanticheskoj blizosti termov // Vestnik YUrGU. Seriya: Vychislitel'naya matematika i informatika. 2017. T. 6, S. 73-83.

ВАСИЛЬЕВ Владимир Иванович, доктор технических наук, профессор кафедры вычислительной техники и защиты информации ФГБОУ ВО «Уфимский государственный авиационный технический университет», Россия, 450008, г. Уфа, ул. К.Маркса, 12. E-mail: vasilyev@ugatu.ac.ru

ВУЛЬФИН Алексей Михайлович, кандидат технических наук, доцент кафедры вычислительной техники и защиты информации, ФГБОУ ВО «Уфимский государственный авиационный технический университет», Россия, 450008, г. Уфа, ул. К.Маркса, 12. E-mail: vulfin.alexey@gmail.com

КИРИЛЛОВА Анастасия Дмитриевна, аспирант кафедры вычислительной техники и защиты информации, ФГБОУ ВО «Уфимский государственный авиационный технический университет», Россия, 450008, г. Уфа, ул. К.Маркса, 12. E-mail: kirillova.andm@gmail.com

НИКОНОВ Андрей Владимирович, аспирант кафедры вычислительной техники и защиты информации, ФГБОУ ВО «Уфимский государственный авиационный технический университет», Россия, 450008, г. Уфа, ул. К.Маркса, 12. E-mail: nikonovandrey1994@gmail.com

VASILYEV Vladimir, Dr. Sc. (Eng.), Professor of the Department «Computer Engineering and Information Security», Ufa State Aviation Technical University». 12, K. Marx Str., Ufa, 450008, Russia. E-mail: vasilyev@ugatu.ac.ru

VULFIN Alexey, PhD (Cand. of Sc.) Ass. Professor of the Department «Computer Engineering and Information Security», Ufa State Aviation Technical University. 12, K. Marx Str., Ufa, 450008, Russia. E-mail: vulfin.alexey@gmail.com

KIRILLOVA Anastasiya, Postgrad. Student of the Department «Computer Engineering and Information Security», Ufa State Aviation Technical University. 12, K. Marx Str., Ufa, 450008, Russia. E-mail: kirillova.andm@gmail.com

NIKONOV Andrey, Postgrad. Student of the Department «Computer Engineering and Information Security», Ufa State Aviation Technical University. 12, K. Marx Str., Ufa, 450008, Russia. E-mail: nikonovandrey1994@gmail.com