УДК 004.056 **DOI** 10.15622/ia.24.1.6

В.Е. Боровков, П.Г. Ключарёв, Д.И. Денисенко

МЕТОДИКА ОЦЕНИВАНИЯ РЕЗУЛЬТАТИВНОСТИ ФУНКЦИОНИРОВАНИЯ СИСТЕМ ОБНАРУЖЕНИЯ ВЕБ-БЭКДОРОВ

Боровков В.Е., Ключарёв П.Г., Денисенко Д.И. **Методика оценивания результативности функционирования систем обнаружения веб-бэкдоров.**

Аннотация. В настоящее время наблюдается значительный рост инцидентов информационной безопасности, связанных с атаками на веб-ресурсы. Получение несанкционированного доступа к веб-ресурсам остается одним из основных методов проникновения в корпоративные сети организаций и расширения возможностей злоумышленников. В связи с этим множество исследований направлено на разработку систем обнаружения веб-бэкдоров (СОВБ), однако существует задача оценивания результативности функционирования данных систем. Цель данного исследования заключается в разработке объективного подхода для оценки результативности функционирования СОВБ. В данной работе было установлено, что объективно результативность СОВБ проявляется в процессе их использования, поэтому тестирование таких систем необходимо проводить в условиях, максимально приближенных к реальным. В связи с этим в статье предложена методика оценивания результативности функционирования СОВБ. Она основана на расчете трех групп частных показателей, характеризующих действенность, ресурсоемкость и оперативность работы системы обнаружения, а также вычислении обобщенного показателя результативности. На основе анализа исследований в данной области была составлена классификация веб-бэкдоров, встраиваемых злоумышленником в исходный код веб-приложений. Эта классификация используется при формировании тестовых наборов данных для вычисления частных показателей действенности. Разработанная методика применима для СОВБ, которые работают на основе анализа исходного кода веб-страниц. Также для ее использования необходим ряд исходных данных, таких как допустимые предельные ошибки частных показателей действенности и вероятность нахождения их в доверительном интервале, а также весовые коэффициенты частных показателей действенности, которые подбираются экспертными методами. Данная работа может быть полезной для специалистов и исследователей в области информационной безопасности, которые хотят проводить объективную оценку своих СОВБ.

Ключевые слова: кибербезопасность, веб-уязвимости, веб-бэкдоры, веб-шеллы, машинное обучение, методы и средства тестирования.

1. Введение. В настоящее время наблюдается увеличение числа компьютерных инцидентов, включая успешные атаки на веб-ресурсы организаций. Как пишут эксперты из компании Positive Technologies [1], захват злоумышленником веб-сайта приводит к расширению его возможностей – от искажения сайта до полной компрометации сети [2]. Так в 2022 году инциденты с веб-ресурсами приводили к нарушениям деятельности организаций в 53% случаях [1], а в 2023 году число атак на веб-ресурсы возросло на 40% по сравнению с аналогичным периодом 2022 года [3].

Тенденция указывает на то, что кибератаки на веб-приложения будут расти как по количеству, так и по уровню сложности. В связи с этим кибербезопасность должна постоянно развиваться и усовершенствоваться, чтобы обеспечить их надежную защиту. Небрежное отношение к обеспечению безопасности веб-приложений может привести к утечке конфиденциальных данных, а также к финансовым и репутационным потерям для организации.

Веб-сайт зачастую не является конечной целью злоумышленника, он может использоваться для проведения атак во внутренней сети организации, а для «закрепления» на веб-ресурсе часто используются веб-бэкдоры. Также в работе [4] было выявлено, что многие исследователи при разработке интеллектуальных методов защиты веб-приложений от уязвимостей, а также веб-бэкдоров, не проводили проверку своих результатов в реальных условиях. Отсюда следует, что результаты их проверки на собственных тестовых наборах могут быть необъективными. Тем самым у специалистов по информационной безопасности (ИБ) может возникнуть вопрос о выборе наиболее результативной системы обнаружения вэб-бэкдоров. Согласно [5] под «результативностью» будем понимать степень реализации запланированной деятельности и достижения запланированных результатов. В контексте систем обнаружения веб-бэкдоров (СОВБ) результативность проявляется в их способности обнаруживать веб-бэкдоры.

Настоящая статья построена следующим образом: в разделе 2 рассмотрены существующие методики тестирования систем обнаружения и защиты от вредоносного программного обеспечения (ВПО); в разделе 3 представлен пример влияния различных наборов тестовых данных на результаты оценивания СОВБ; в разделах 4-6 предлагается методика оценивания результативности функционирования СОВБ, апробация которой представлена в разделе 7.

2. Существующие методики тестирования систем обнаружения и защиты от вредоносного программного обеспечения. *Веб-бэкдор* является скрытым механизмом, который позволяет злоумышленнику получать удаленный несанкционированный доступ к веб-серверу для выполнения различных операций. Следует отметить, что существует множество разновидностей веб-бэкдоров, которые могут быть встроены в веб-приложения, а также в веб-сервера (например, *Apache-бэкдоры*, *Nginx*-бэкдоры [6]) и т.д. Хотя некоторые уязвимости веб-приложений или веб-серверов могут представлять собой веб-бэкдоры, в данной статье под последними понимается вредоносные сценарии (такие как веб-шеллы, веб-загрузчики и прочее [4]),

встраиваемые злоумышленником в исходный код веб-приложения на интерпретируемом языке программирования. Но для встраивания таких бэкдоров в веб-приложение злоумышленник должен иметь возможность изменения файлов на сервере. Это может быть достигнуто различными способами, в том числе через уязвимости веб-приложения [7]. В любом случае веб-бэкдор будет являться ВПО.

Существует множество факторов, влияющих на результаты тестирования систем обнаружения и защиты от ВПО. Различные методы тестирования могут давать разные результаты, так как они могут не учитывать все возможные варианты атак. Кроме того, качество обновлений и скорость реакции на новые угрозы также могут влиять на результаты тестирования. Наконец, каждый пользователь имеет свои собственные потребности и предпочтения, которые могут повлиять на выбор лучшей системы обнаружения и защиты от ВПО для его конкретных нужд. Однако в отношении СОВБ не было обнаружено четкой методики оценивания их результативности, исходя из анализа документов, представленных в [4, 8].

В настоящее время существуют следующие методики тестирования систем обнаружения и защиты от ВПО:

1) *On-demand scan (ODS).* Представляет собой статический анализ ВПО с помощью систем защиты [9]. Статический анализ подразумевает проверку ВПО без его фактического выполнения; например, система обнаружения может применять сигнатурный метод для выявления ВПО. В этой методике предполагается, что результативность системы обнаружения определяется количеством обнаруженных ВПО в процессе статического анализа. Для проведения качественного тестирования важно использовать широкий спектр вредоносных программ, который состоит из более чем тысячи файлов и документов. Некоторые специализированные организации предлагают такие коллекции. Однако использование только этих тестов не всегда дают объективную оценку результативности из-за возможности злоумышленников применять обфускацию и шифрование, а также комбинировать их с другими методами, которые позволяют обходить статический анализ.

2) *Динамическое тестирование (Тестирование с запуском).* Суть данной методики заключается в запуске вредоносных средств в виртуальной среде и их анализе системами защиты (также называется *эвристическим анализом*). Вместе со статическим анализом этот подход может быть результативным, но у него есть существенный недостаток: вредоносные программы могут представлять только часть цепочки атаки, и для их полноценной

работы могут потребоваться дополнительные условия и параметры, которые виртуальная среда может не предоставить. Также ВПО способно определять среду выполнения и не реализовывать свои вредоносные функции.

3) *Real-world test (RW)* [10]. Данная методика является наиболее сложной, так как представляет собой моделирование полного цикла заражения реальной системы и анализ реакции системами защиты. Такие тесты позволяют выявить различные проблемы, с которыми программное обеспечение безопасности может столкнуться при работе в реальных условиях с реальными угрозами. Однако главным недостатком является сложность создания лабораторной среды. Для достоверных результатов тестирования лабораториям приходится использовать физические компьютеры, а не виртуальные машины, что требует постоянного восстановления систем после каждого запуска ВПО.

Данные методики тестирования наиболее распространены и охватывают широкий спектр вредоносных программ [11], что позволяет проверить большое количество систем обнаружения и защиты от ВПО. Их сравнительная характеристика представлена в таблице 1.

Таблица 1. Сравнительная характеристика методов тестирования систем обнаружения и защиты от ВПО

№ п/п	Наименование методики	Достоинства	Недостатки
1	ODS-тестирование	– относительная простота реализации	– необходимо иметь большую базу ВПО – не позволяет оценивать системы обнаружения и защиты, основанные на эвристическом анализе
2	Динамическое тестирование	– позволяет оценивать системы обнаружения и защиты, основанные на поведенческом анализе	– необходимость создания виртуальной среды – виртуальная среда может не в полной степени создать условия для успешного развертывания ВПО
3	RW-тестирование	– позволяет выявить недостатки используемых в системах обнаружения и защиты алгоритмов статического и эвристического анализа – позволяет имитировать различные виды атак	– сложность создания лабораторной среды для проведения тестирования

Кроме того, имеются другие методы, которые направлены на проверку производительности, скорости реакции и других характеристик систем обнаружения и защиты от вредоносных программ. Однако, если необходимо протестировать систему, работающую в узконаправленной области (например, обнаружение веб-бэкдоров), требуются специализированные тесты.

3. Реакция СОВБ на различные наборы веб-бэкдоров. Существуют разные методы для выявления веб-бэкдоров, которые основываются на различных принципах, включая анализ файлов сайта, логов веб-сервера, HTTP-трафика и другие. При тестировании СОВБ необходимо учитывать эти принципы. В данной статье рассмотрим СОВБ, работающих по принципу анализа исходного кода веб-приложений. Многие методы обнаружения используют именно этот подход [8]. Одной из основных проблем при тестировании таких систем является недостаток качественных или полных наборов тестовых данных, а также игнорирование некоторыми исследователями возможных модификаций веб-бэкдоров, которые позволяют обойти системы защиты и обнаружения [4]. Для подтверждения этого, в качестве примера был рассмотрен один из методов обнаружения веб-бэкдоров, основанный на сверточной нейронной сети [12]. Алгоритм [13] производит анализ исходных кодов набора PHP-страниц, чтобы получить последовательность инструкций операционных кодов (опкодов). Затем используется *Word2vec* [14] для получения векторной карты для массивов опкодов и для массивов биграмм опкодов. Наконец, они служат двумя входами сверточной нейронной сети, которая осуществляет обнаружение. Пример того, как выглядят опкоды языка PHP, представлен на рисунке 1. Похожие идеи использовали также исследователи в работе [15].

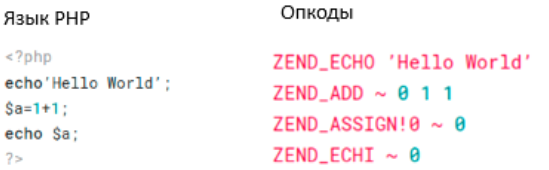

Рис. 1. Опкоды языка PHP

Исследователь утверждает, что алгоритм достигает точности (*accuracy*) 98,4%, что подтверждается нами при проверке на тестовом наборе данных, который использовался исследователем для тестирования.

Были выбраны 6 различных веб-бэкдоров, которые подверглись анализу с помощью алгоритма обнаружения. Среди них следующие веб-бэкдоры:

1) Веб-загрузчик. Представляет собой обычную форму загрузки файлов через два POST параметра: первый параметр задает имя файла, который необходимо создать, а второй – текст в кодировке base64, который нужно внести в этой файл.

2) Веб-загрузчик. Загрузка файлов через два POST параметра: первый параметр – директория сохранения файла, второй – сам файл.

3) Веб-шелл, который принимает через POST параметр закодированную в *Base64* команду и выполняет через функцию *eval(system(“command”)).*

4) Веб-шелл, который принимает через GET параметр команду и выполняет через функцию *system(“command”).*

5) Веб-бэкдор, принимающий на вход IP-адрес и порт атакующего и реализующего *Reverse Shell* [16] к злоумышленнику.

6) Большой веб-шелл, который имеет графический интерфейс и выполняет различные операции. Его код представлен на *Github*[17].

Очевидно, что различных вариаций вэб-бэкдоров может быть множество, в данном случае были выбраны простые варианты, которые часто используются злоумышленниками.

В первом случае веб-бэкдоры были представлены в виде отдельных файлов и использовались в качестве входных данных для алгоритма обнаружения с целью прогнозирования. В итоге алгоритм допустил ошибку лишь в одном из шести случаев (для третьего варианта), определив его как легитимный файл (файл, не содержащий веб-бэкдора).

Во втором случае эти же веб-бэкдоры встраивались в легитимный файл. Тем самым легитимные файлы становились носителями вэб-бэкдоров, и должны помечаться СОВБ, как вэб-бэкдоры. Для примера был взят файл *wp-login.php* из популярного фреймфорка *WordPress* [18]. Важным условием являлось то, чтобы веб-страница и веб-бэкдор работали корректно, иначе алгоритм не сможет вычислить операционные коды скриптов. В результате из шести веб-бэкдоров алгоритм верно определил только в одном случае – для загрузчика 1. В остальных случаях алгоритм принял неправильное решение, указав, что это легитимные файлы.

Результаты классификации алгоритмом веб-бэкдоров представлены в таблице 2.

Таблица 2. Воздействие модификации веб-бэкдоров на результативность алгоритма обнаружения

	Номер веб-бэкдора					
	1	2	3	4	5	6
Веб-бэкдор отдельным файлом	+	+	-	+	+	+
Веб-бэкдор встраивается в легитимный файл	+	-	-	-	-	-

Отсюда можно сделать вывод, что незначительная манипуляция с кодом веб-бэкдора может существенно повлиять на результаты алгоритмов обнаружения. В данном случае веб-бэкдоры были внедрены в легитимные файлы сайта, а не являлись отдельными файлами. Также конкретные способы обхода средств обнаружения можно увидеть в документах [19 – 20]. Это поднимает вопрос о том, насколько точность алгоритма, оцененная на тестовом наборе данных (98.4%), отражает его способность обнаруживать 98.4% реально существующих веб-бэкдоров. Метрики точности (*accuracy, precision*) и полноты (*recall*) алгоритмов обнаружения зависят от конкретных наборов данных и поэтому не всегда являются надежными показателями способности алгоритма обнаруживать веб-бэкдоры, поскольку исследование возможных вариантов веб-бэкдоров не проводилось, а сам алгоритм (или СОВБ) не тестировался в реальных условиях.

***Примечание**. В англоязычной литературе метрики precision и accuracy имеют различные значения, определяемые формулами (5) и (6) соответственно, при этом перевод у них одинаковый – «точность», также accuracy иногда называют «меткостью».*

Тем самым для ответа на следующие вопросы: «как сравнить несколько систем обнаружения веб-бэкдоров и выбрать наиболее результативный?» и «какие наборы данных следует использовать для объективного расчета показателей результативности?», необходимо разработать методику, которую могут использовать специалисты и исследователи в области ИБ для объективной оценки результативности функционирования СОВБ.

4. Постановка задачи. Цель работы – разработка объективного подхода для оценки результативности функционирования СОВБ. Для этого требуется разработать методику оценивания результативности функционирования СОВБ. Ввиду того, что СОВБ могут работать по различным принципам (анализ HTTP-пакетов, анализ логов системы, анализа исходного кода веб-страниц и др.) выделим исходное ограничение, что оцениваемые системы работают на основе анализа

исходного кода веб-страниц. Также предполагается, что изначально известна среда функционирования СОВБ: операционная система (ОС) и аппаратные характеристики устройства, где развернута СОВБ, такие как объем оперативной памяти, модель процессора, объем видеопамяти (в случае использования СОВБ видеокарты), а также известна нагрузка на СОВБ (средний предполагаемый объем анализируемых файлов).

Согласно базовым понятиям теории и методологии внешнего проектирования целенаправленных процессов и целеустремленных систем [21] можно определить, что объективно результативность СОВБ проявляется в процессе их использования. *Результативность,* является основным свойством системы. Также к ряду основных свойств могут относить *ресурсоемкость* (затраты ресурсов системой) и *оперативность* (затраты времени, скорость реакции).

Количественной мерой результативности может выступать обобщённый показатель результативности системы, который характеризует результат ее функционирования при заданных характеристиках ее состояния и определённых внешних воздействиях. Он может быть представлен с помощью формулы (1):

$$E = \sum_{i=1}^{m} e_i w_i , \qquad (1)$$

где E – обобщенный показатель результативности СОВБ, e_i – значение i-го частного показателя действенности СОВБ, при этом $0 \le e_i \le 1$, w_i – весовой коэффициент i-го частного показателя и $\sum_{i=1}^{m} w_i = 1$, $i = \overline{1, m}$ – количество частных показателей действенности.

Выбор и расчет частных показателей действенности зависит от специфики работы системы. Для получения значений частных показателей СОВБ можно применять метод, основанный на проведении испытаний на тестовом стенде, который максимально приближен к реальным условиям эксплуатации системы [22].

Оценивание возможно только в системе, которой предъявлены требования. Целью оценивания является выработка суждения о СОВБ на основе полученных (измеренных) показателей. Такое суждение формируется с помощью определенных правил и принципов, которые выражены в форме критериев оценивания. Обоснованный выбор наиболее подходящих решений и средств осуществляется посредством

сравнения полученных в ходе испытаний результатов с заданными требованиями.

Оценивание результативности СОВБ можно разбить на следующие этапы:

1. Формирование критериев на основе требований к СОВБ. Требования задаются исследователем.
2. Составление сценария проведения эксперимента исследователем.
3. Создание лабораторной среды для проведения эксперимента, подготовка тестовых наборов данных исследователем.
4. Расчет (получение) значений частных показателей.
5. Анализ полученных значений частных показателей и принятие решения о пригодности СОВБ.
6. Получение значения обобщённого показателя результативности.
7. Анализ полученных результатов, сравнение СОВБ, принятие решений исследователем.

Далее рассмотрим этапы оценивания более подробно.

Этап 1.

Системы, предназначенные для обнаружения веб-бэкдоров, могут играть важную роль в обеспечении безопасности веб-приложений. Однако такие системы должны не только выполнять свою основную функцию, но и соответствовать определенным требованиям:

1) не нарушать работоспособность веб-приложений;
2) осуществлять обнаружение веб-бэкдоров за адекватное время;
3) не становиться основным потребителем ресурсов на сервере и не приводить к снижению производительности веб-приложений.

Для того, чтобы учесть эти требования, установим три группы частных показателей, каждая из которых отражает действенность, ресурсоемкость (затраты ресурсов) и оперативность (затраты времени) функционирования СОВБ. Все три группы частных показателей будем использовать, только для определения пригодности СОВБ на этапе 5. Для этапов 6-7 – вычисление интегрального показателя результативности, сравнение и выбор подходящей СОВБ – будут использоваться только частые показатели действенности.

Определим вектор частных показателей, который требуется вычислить (получить), как $Y_{\langle n \rangle} = \langle y_1, y_2, ..., y_n \rangle = \langle e_1, e_2, ..., e_{n1},$

$r_1, r_2, ..., r_{n2}, t_1, t_2, ..., t_{n3}\rangle$, где $e_1, e_2, ..., e_{n1}$ – показатели, характеризующие действенность, $r_1, r_2, ..., r_{n2}$ – показатели, характеризующие ресурсоемкость, $t_1, t_2, ..., t_{n3}$ – показатели, характеризующие оперативность функционирования СОВБ, а $n = n1 + n2 + n3$ – общее количество частных показателей.

При вычислении частных показателей действенности получаются точечные значения. Для того чтобы делать содержательные выводы, необходимо находить доверительный интервал, т.е. интервал, который с заданной вероятностью накрывает значение частного показателя. К факторам, влияющим на ширину доверительного интервала, относятся размер выборки, изменчивость выборки и доверительная вероятность нахождения показателя в данном интервале. Поэтому сформируем критерий достаточности тестового набора данных (наборы зараженных файлов и легитимных файлов) для оценки частных показателей действенности СОВБ (S). Пусть $\Delta_e^d = \langle \Delta_{e1}^d, \Delta_{e2}^d, ..., \Delta_{n1}^d \rangle$ – вектор допустимых предельных ошибок вычисления частных показателей действенности $e_1, e_2, ..., e_{n1}$, а $\Delta_e = \langle \Delta_{e1}, \Delta_{e2}, ..., \Delta_{en1} \rangle$ – вектор предельных ошибок вычисления частных показателей $e_1, e_2, ..., e_{n1}$, полученный после проведения эксперимента, исходя из объема тестового набора данных. Тогда критерий будет выглядеть следующим образом:

$$S : (\Delta_{e_i} \le \Delta_{e_i}^d) \cong U, i = 1, .., n1, \tag{2}$$

где $\cong$ – знак равносильности высказываний, U – достоверное событие (истинное высказывание), $n1$ – количество частных показателей действенности.

Вектор допустимых предельных ошибок Δ_e^d и доверительная вероятность являются исходными данными. Они задаются экспертными методами.

На основании требований к системе обнаружения можно выделить область допустимых значений частных показателей. Пусть $\{y_i^d\}$ – множество (область) допустимых значений показателя y_i. Или в векторной форме $\{Y_{\langle n \rangle}^d\} = \{\langle y_1^d, y_2^d, ..., y_n^d \rangle\}$. Тогда критерий пригодности для СОВБ (G) будет выглядеть следующим образом [21]:

$$G:(Y_{\langle n\rangle}\in\left\{Y^d_{\langle n\rangle}\right\})\cong U, \tag{3}$$

где $\left\{Y^d_{\langle n\rangle}\right\}$ – область допустимых значений частных показателей.

Для вычисления обобщенного показателя результативности по формуле (1) также необходимо задать вектор весовых коэффициентов частных показателей действенности $W_e=\left\langle w_{e_1}, w_{e2},..., w_{en1}\right\rangle$. Весовые коэффициенты W_e также задаются изначально экспертными методами на основе приоритетности тех или иных частных показателей действенности.

Этап 2.

Для получения значений показателей $y_1, y_2,..., y_n$ СОВБ необходимо разработать сценарий проведения эксперимента. В сценарии поясняется, как и каким образом будут получены значения частных показателей.

Этап 3.

На основе сценария проведения эксперимента создается лабораторный стенд, а также подготавливается наборов тестовых данных – легитимных файлов и веб-бэкдоров. Формирование наборов данных является одним из основных шагов для вычисления частных показателей действенности. Особое внимание уделяется формированию набора тестовых данных для различных вариантов веб-бэкдоров: $D^1_{m1}, D^2_{m2},..., D^k_{mk}$, где $D^i_{mi}=\left\{d^i_1, d^i_2,..., d^i_{mi}\right\}$, i указывает на вид веб-бэкдора, mi – количество различных вариантов для веб-бэкдора i-го вида.

Этапы 4, 5.

На данных этапах проводятся испытания и вычисляются частные показатели на основе проведенного эксперимента. Также рассчитывается вектор предельных ошибок частных показателей действенности Δ_e. Если критерий достаточности набора тестовых данных S (2) не выполняется, то принимается решение на формирование большего набора тестовых данных. Затем эксперимент проводится повторно. Если же критерий S выполняется, то на основе критерия пригодности G (3) исключаются те СОВБ, которые не соответствуют требованиям.

Этапы 6, 7.

С использованием весовых коэффициентов W_e рассчитывается результативность с помощью формулы (1) для каждой СОВБ, участвующей в исследовании и удовлетворяющей критериям S и G. Затем принимается решение о выборе наилучшей системы обнаружения, которая имеет наибольшее значение результативности.

Как видно из перечисленных этапов, процесс оценивания основывается на вычислении частных показателей $Y_{\langle n \rangle} = \langle y_1, y_2, ..., y_n \rangle$ с использованием тестовых наборов данных в ходе проведения эксперимента, а также на вычислении результативности E. Согласно уравнению (1) $0 \le E \le 1$, и чем выше значение E, тем результативнее работает СОВБ.

Далее сформируем вектор частных показателей для методики оценивания результативности СОВБ и рассмотрим процесс составления тестовых наборов данных для расчета частных показателей.

5. Формирование вектора частных показателей. Для того, чтобы всецело охватить совокупность требований к системам обнаружения, в предыдущем разделе был введен вектор $Y_{\langle n \rangle}$. Действенность системы обнаружения веб-бэкдоров отображена в элементах $e_1, e_2, ..., e_{n1}$. В данной работе были выбраны три показателя, представленные в выражениях (4-6). В их основе лежат следующие базовые переменные, которые присущи бинарному классификатору:

TP – истинно положительные классификации, т.е. случаи, когда система правильно обнаружила веб-бэкдор;

TN – истинно отрицательные классификации, т.е. случаи, когда система правильно определила отсутствие веб-бэкдора;

FP – ложноположительные классификации, т.е. случаи, когда система неправильно определила наличие веб-бэкдора (ошибка первого рода);

FN – ложноотрицательные классификации, т.е. случаи, когда система неправильно определила отсутствие веб-бэкдора (ошибка второго рода).

$$recall : e_1 = Re, \text{ где } Re = \frac{TP}{TP + FN}, \tag{4}$$

$$precision : e_2 = Pr, \text{ где } Pr = \frac{TP}{TP + FP}, \tag{5}$$

$$accuracy: e_3 = Ac, \text{где } Ac = \frac{TP + TN}{TP + TN + FP + FN}. \quad (6)$$

Частный показатель $e_1 = Re$ отражает долю веб-бэкдоров, обнаруженных СОВБ, от общего числа веб-бэкдоров в тестовом наборе данных. При этом $0 \le e_1 \le 1$.

Частный показатель $e_2 = Pr$ отражает долю объектов, классифицированных СОВБ как веб-бэкдоры, и при этом действительно являющимися таковыми. При этом $0 \le e_2 \le 1$.

Частный показатель $e_3 = Ac$ отражает долю правильных классификаций СОВБ. При этом $0 \le e_3 \le 1$.

Эксперимент проводятся на наборе легитимных файлов, а также на наборах веб-бэкдоров $D^1_{m1}, D^2_{m2}, ..., D^k_{mk}$, которые будут рассмотрены подробнее в следующем разделе.

Следующую группу показателей, характеризующую затраты ресурсов на работу СОВБ, получим из загруженности оперативной памяти, видеокарты и процессора на выполнение операций:

$$r_1 = V_{\text{оп}}, r_2 = L_{\text{цп}}, r_3 = V_{\text{гп}},$$

где $V_{\text{оп}}$ – максимальный объем оперативной памяти, потребляемый системой (в Мбайтах/Гбайтах), $L_{\text{цп}}$ – максимальная загруженность процессора данной системой (в процентах), $V_{\text{гп}}$ – максимальный объем видеопамяти, используемый системой (в Мбайтах/Гбайтах), за время функционирования СОВБ. Частный показатель r_3 актуален для систем, использующих машинное обучение для выполнения обнаружения.

Также немаловажным аспектом работы СОВБ является третья группа показателей, которая характеризует оперативность. При исследовании СОВБ, основанных на анализе содержимого веб-файлов, можно использовать один показатель $t_1 = \tau$, где τ – среднее время анализа содержимого файла. Его можно вычислить, разделив общее время анализа файлов на количество проанализированных файлов.

Следует отметить, что показатели ресурсоемкости ($r_1, r_2, ..., r_{n2}$) и оперативности ($t_1, t_2, ..., t_{n3}$) зависят от технических особенностей

программной среды, где развернута СОВБ, а также от нагрузки на СОВБ. Поэтому при расчете данных показателей важно указывать использованные технические особенности среды (версия ОС, количество оперативной памяти, видеопамяти, процессор) и объем нагрузки на СОВБ (количество проанализированных файлов).

Таким образом, вектор частных показателей для разрабатываемой методики будет иметь следующий вид: $Y_{\langle 7\rangle} = \langle e_1, e_2, e_3, r_1, r_2, r_3, t_1\rangle = \langle Re, Pr, Ac, V_{\text{оп}}, L_{\text{цп}}, V_{\text{гп}}, \tau\rangle$.

6. Подготовка тестовых наборов данных для расчета частных показателей

6.1. Классификация веб-бэкдоров, встраиваемых в исходный код веб-приложения. Одним из важных этапов при оценивании СОВБ является подготовка тестовых наборов.

Тестовые наборы данных для получения значений частных показателей ресурсоемкости и оперативности формируются за счет исходных данных (объем файлов анализируемого веб-приложения). Эти показатели характеризуют затраты ресурсов (оперативной памяти, процессора, видеокарты) и среднее время анализа файла. Поэтому наличие или отсутствие веб-бэкдоров в тестовых наборах данных не влияет на значение данных показателей.

Чтобы получить объективные значения частных показателей действенности, необходимо использовать различные варианты вэб-бэкдоров, которые могут быть внедрены злоумышленниками в исходный код веб-приложений. Для этого были изучены тестовые наборы, упомянутые в работах [15, 23 – 26]. Анализ данных тестовых наборов данных показал, что веб-бэкдоры, встроенные в исходный код веб-приложений, могут выполнять различные функции и делятся на 4 вида: выполнение команд операционной системы; выполнение команд языка программирования веб-приложения; загрузка файлов (других веб-бэкдоров); выполнение специализированных операций (к примеру выполнение Reverse Shell к злоумышленнику, создание веб-прокси и т.д.). Код веб-бэкдора не всегда содержится в одном файле, также злоумышленник может использовать различные методы для его сокрытия, такие как шифрование, обфускация или внедрение в легитимные файлы. Для выполнения различных функций злоумышленник может передавать разные значения в параметрах HTTP-запросов или не передавать ничего, если в коде веб-бэкдора уже содержатся все необходимые значения.

На основе этого анализа была разработана классификация веб-бэкдоров (рисунок 2).

Рис. 2. Классификация веб-бэкдоров для создания тестовых наборов данных

Каждый такой веб-бэкдор можно описать с помощью пяти независимых характеристик. К примеру, веб-бэкдор может обладать следующими свойствами:

1. дает возможность выполнять команды операционной системы (блок 1.2, рисунок 2);
2. код веб-бэкдора находится в нескольких файлах (блок 2.2);
3. веб-бэкдору передаются необходимые параметры в теле запроса HTTP (блок 3.1);
4. в коде веб-бэкдора используется обфускация (блок 4.1);
5. код веб-бэкдора не встраивается в легитимные файлы (блок 5.1).

Путем перебора всех возможных вариантов получается 64 вида различных веб-бэкдоров.

В рамках отдельной характеристики возможно объединение, например, веб-бэкдор может выполнять как команды языка программирования веб-приложения (1.1), так и операционной

системы (1.2). Однако в контексте тестовых наборов целесообразно не использовать такое объединение.

При формировании наборов данных воспользуемся выборкой один к одному: один веб-бэкдор к одному легитимному файлу. Это необходимо для того, чтобы одинаково учитывать реакцию СОВБ как на веб-бэкдоры, так и на легитимные файлы при последующем расчете показателей действенности, так как показатель $e_з = Ac$ (6) бесполезен в задачах с неравными классами [27]. Тем самым в выборке будет $N_{вб}$ веб-бэкдоров и $N_л$ легитимных файлов, при этом $N_{вб} = N_л = N$. В свою очередь, исходя из классификации, представленной на рисунке 2, пространство веб-бэкдоров можно разделить на 64 группы. Набор веб-бэкдоров будем формировать на основе типической выборки, пропорциональной объему типических групп, предполагая, что веб-бэкдоры равнозначно распределены по этим группам (имеют одинаковые объемы групп). Таким образом, при отборе существует два набора – веб-бэкдоры и легитимные файлы – одинаковые по объему. В свою очередь веб-бэкдоры разделяются на 64 группы. Например, если использовать отбор из 256 элементов – 128 из них легитимные элементы, а 128 – веб-бэкдоры. При этом вэб-бэкдоры представлены набором $D^1_{m1}, D^2_{m2}, ..., D^k_{mk}$, где $m1, m2, ..., mk = 2$ (по две реализации на каждый вид веб-бэкдора), а $k = 64$, т.е. $D^1_2, D^2_2, ..., D^{64}_2$, где $D^i_2 = \left\{d^i_1, d^i_2\right\}$.

Был создан набор PHP-файлов, который можно использовать для расчета показателей действенности СОВБ. Этот набор файлов доступен в репозитории на *Github* [28]. При принятии решения о выборе языка программирования был учтен факт, что PHP является наиболее распространенным языком для создания различных типов веб-приложений [29]. Для имитации веб-приложения был использован фреймворк *WordPress*. Также в репозитории содержатся примеры команд для проверки работоспособности веб-бэкдоров и создания запросов к ним. Так в начале подраздела был представлен веб-бэкдор с определёнными свойствами. Его реализацию можно увидеть на рисунке 3. Веб-бэкдор состоит из двух файлов: *25.php* и *25.1.php*. Оба этих файла обфусцированы, что затрудняет их анализ. Для реализации логики используется метод *goto*, который позволяет перескакивать между инструкциями. Обфускация достигается за счет применения множества неочевидных символов, сокращений и необычных названий переменных, что делает код менее читаемым и усложняет его понимание.

Файл 25.php

```
<?php
/*   ____________________________________________
     |  Obfuscated by YAK Pro - Php Obfuscator  2.0.14  |
     |    GitHub: https://github.com/pk-fr/yakpro-po    |
     |__________________________________________________|
*/
goto jhXrD;
gyobV: require "\x77\x70\x2d\160\x6f\x6c\x69\x6e\56\x70\x68\x70";
goto TH36a; jhXrD: require "\62\65\56\x31\56\160\150\x70"; goto iM_oz;
TH36a: wp_run($_POST["\167\x70\55\160\x6f\x6c\151\x6e"]); goto T1S31;
iM_oz: sis_create(); goto gyobV;
T1S31: unlink("\167\x70\x2d\x70\157\x6c\x69\x6e\x2e\x70\150\160");
```

Файл 25.1.php

```
<?php function sis_create(){$xafq_0=fopen('wp-polin.php','w');
$xemy_1="<?php function ";$iosx_2="wp_run(\$bd){pas";$eduj_3="sth";
$syeg_4="ru(\$bd);}?>";fputs($xafq_0,$xemy_1.$iosx_2.$eduj_3.$syeg_4);}?>
```

Рис. 3. Пример реализации веб-бэкдора

Таким образом, при разработке сценария эксперимента расчет можно осуществлять в два этапа. На первом этапе следует определить значения частных показателей оперативности и ресурсоемкости для всего доступного объема скриптов, основываясь на среднем предполагаемом объеме анализируемых файлов, независимо от наличия или отсутствия в них бэкдора. На втором этапе проводятся расчеты частных показателей действенности. Данные для анализа представляют собой легитимные файлы и веб-бэкдоры, сформированные по указанной выше выборке.

6.2. Определение требуемого объема выборки и предельных ошибок частных показателей действенности. Для получения объективной оценки тестирования рекомендуется расширить набор, который используется для расчета частных показателей действенности.

Необходимо отметить, что, согласно выборке, которая была предложена в подразделе 6.1, а также по определению переменных TP, TN, FP, FN из раздела 5, следует, что

$$TP + FN = N_{\text{вб}} = N, \tag{7}$$

$$TN + FP = N_{\text{л}} = N\,. \tag{8}$$

Отсюда получаем, что результат анализа легитимных объектов не влияет на значения TP и FN. Обозначим $w_{TP} = \frac{TP}{N_{\text{вб}}} = \frac{TP}{N}$, $w_{FN} = \frac{FN}{N_{\text{вб}}} = \frac{FN}{N}$ – соответственно доля TP и FN относительно общего числа веб-бэкдоров в выборке. Аналогично результат анализа веб-бэкдоров не влияет на значения TN и FP. Обозначим, $w_{TN} = \frac{TN}{N_{\text{л}}} = = \frac{TN}{N}$, $w_{FP} = \frac{FP}{N_{\text{л}}} = \frac{FP}{N}$ – соответственно доля TN и FP относительно общего числа легитимных объектов в выборке.

За счет вычисления доверительных интервалов можно с требуемой вероятностью ограничить значения $w^*_{TP}, w^*_{TN}, w^*_{FP}, w^*_{FN}$ – истинные значения долей TP, TN, FP, FN, соответственно.

Вычислим, сколько объектов необходимо исследовать, чтобы с вероятностью $P = 0.997$ и ошибкой не более $\Delta = 0.05$ определить значения $w^*_{TP}, w^*_{TN}, w^*_{FP}, w^*_{FN}$.

Определение необходимого количества веб-бэкдоров для оценки w^*_{TP} *(*w^*_{FN}*).*

Так как набор вэб-бэкдоров формируются за счет типической выборки, пропорциональной объему групп, то формула для расчета необходимой численности выборки в данном случае будет иметь вид [30]:

$$N_{\text{вб}} = \frac{z^2 \overline{\sigma_w^2}}{\Delta_w^2}, \tag{9}$$

где Δ_w^2 – квадрат предельной ошибки доли, $\overline{\sigma_w^2}$ – средняя из групповых дисперсий типических групп для $w_{TP}(w_{FN})$, z – коэффициент доверия, который определяется на основе табличных значений в зависимости от вероятности P.

Средняя из групповых дисперсий вычисляется по формуле:

$$\overline{\sigma_w^2} = \frac{\sum_{j=1}^{k} \sigma_j^2 N_j}{\sum_{j=1}^{k} N_j}, k = 64, \tag{10}$$

где N_j – количество элементов в j-й типической группе, k – количество типических групп, σ_j^2 – дисперсия выборочной доли в j-й типической группе.

Как упоминалось ранее, существует 64 типических групп (64 вида веб-бэкдоров). Дисперсия выборочной доли в j-й типической группе определяется по формуле [30]:

$$\sigma_j^2 = w_j(1-w_j), \tag{11}$$

где w_j – доля $TP(FN)$ в j-й типической группе.

Так как изначально неизвестен характер реакции СОВБ мы будем полагать дисперсию в каждой группе максимальной. Она достигается при $w_j = 0,5$ и равна $\sigma_j^2 = 0,25$ [31]. $\sum_{j=1}^{64} N_j = N$, а $N_j = N/64$. Тогда $\overline{\sigma_w^2} = 0,25$. Значение z табличное, которое при $P = 0,997$ равно 3.

Таким образом:

$$N_{\text{вб}} = \frac{3^2 \cdot 0,25}{0,05^2} = 900.$$

Так как используется 64 вида веб бэкдоров, то необходимо $\frac{900}{64} = 14,063 \approx 15$ веб-бэкдоров каждого вида (960 веб-бэкдоров): $D_{15}^1, D_{15}^2, ..., D_{15}^{64}$, где $D_{15}^i = \left\{d_1^i, d_2^i, ..., d_{15}^i\right\}$.

Определение необходимого количества легитимных файлов для оценки w_{FP}^* (w_{TN}^*).

Так как легитимные файлы не разделяются по типам, то формула для расчета необходимой выборки в данном случае соответствует собственно-случайному отбору и имеет вид:

$$N_{\text{л}} = \frac{z^2 \cdot \sigma_w^2}{\Delta_w^2}, \tag{12}$$

где σ_w^2 – дисперсия доли в выборке. Она вычисляется по формуле:

$$\sigma_w^2 = w(1-w), \tag{13}$$

где w – доля $FP(TN)$ в выборке легитимных файлов.

Также полагаем, что она максимальна, т.е. $\sigma_w^2 = 0,25$. Получаем:

$$N_{\text{л}} = \frac{3^2 \cdot 0,25}{0,05^2} = 900.$$

Для обеспечения равного количества веб-бэкдоров и легитимных файлов, требуется минимум 1920 элементов (960 веб-бэкдоров и 960 легитимных файлов), чтобы с вероятностью 0.997 и ошибкой не более $\Delta_{wTP} = \Delta_{wTN} = \Delta_{wFP} = \Delta_{wFN} = 0,05$ определить $w_{TP}^*, w_{TN}^*, w_{FP}^*, w_{FN}^*$. Таким образом, необходимо, как минимум, 15 пакетов по 128 элементов каждый. При этом условии будут выполняться следующие неравенства:

$$\begin{gathered} w_{TP} - \Delta_{wTP} \le w_{TP}^* \le w_{TP} + \Delta_{wTP}, \\ w_{TN} - \Delta_{wTN} \le w_{TN}^* \le w_{TN} + \Delta_{wTN}, \\ w_{FP} - \Delta_{wFP} \le w_{FP}^* \le w_{FP} + \Delta_{wFP}, \\ w_{FN} - \Delta_{wFN} \le w_{FN}^* \le w_{FN} + \Delta_{wFN}. \end{gathered} \tag{14}$$

Набора из 15 пакетов будет достаточно, чтобы оценить значения $w_{TP}^*, w_{TN}^*, w_{FP}^*, w_{FN}^*$, с вероятностью 0,997 и ошибкой не более 0,05. Однако если есть возможность предположить дисперсию долей, то возможно уменьшение объема выборки.

Если взять выборку веб-бэкдоров и легитимных файлов, то для w_{TP}^* и w_{FN}^* ошибка будет вычисляться по формуле (15), а для w_{FP}^*

и w_{TN}^{*} по формуле (16). Количество веб-бэкдоров и легитимных файлов равно $N_{\text{вб}} = N_{\text{л}} = N$.

$$\Delta_{w_{TP}} = \Delta_{w_{FN}} = z\sqrt{\frac{\overline{\sigma_w^2}}{N}}, \tag{15}$$

$$\Delta_{w_{TN}} = \Delta_{w_{FP}} = z\sqrt{\frac{\sigma_w^2}{N}}, \tag{16}$$

где σ_w^2 вычисляется по формуле (13), а $\overline{\sigma_w^2}$ по формуле (10).

Для малых объемов выборок, при невозможности вычислить $\overline{\sigma_w^2}$, можно ограничить это значение. Пусть количество веб-бэкдоров $N_{\text{вб}}$ кратно 64 (исходя из предыдущих рассуждений), тем самым в каждой подгруппе будет $\frac{N_{\text{вб}}}{64} = \frac{N}{64} = m$ веб-бэкдоров. Средняя выборочная доля *TP* (для *FN* проводятся аналогичные вычисления) равна:

$$\overline{w_{TP}} = \frac{\sum_{j=1}^{64} w_{TP_j} N_j}{\sum_{j=1}^{64} N_j} = \frac{m\sum_{j=1}^{64} w_{TP_j}}{N} = \frac{\sum_{j=1}^{64} w_{TP_j}}{64}, \tag{17}$$

где w_{TP_j} – доля *TP* в j-й подгруппе, N_j – количество веб-бэкдоров в каждой подгруппе (равно m).

Согласно формулам (10) и (11) получаем:

$$\begin{aligned}\overline{\sigma_w^2} &= \frac{\sum_{j=1}^{64} w_j(1-w_j)}{64} = \frac{\sum_{j=1}^{64} w_{TP_j}(1-w_{TP_j})}{64} = \\ &= \frac{\sum_{j=1}^{64} w_{TP_j}}{64} - \frac{\sum_{j=1}^{64} w_{TP_j}^2}{64} = \overline{w_{TP}} - \frac{\sum_{j=1}^{64} w_{TP_j}^2}{64}.\end{aligned} \tag{18}$$

С другой стороны:

$$\overline{w_{TP}} = \frac{TP}{N} = \frac{\sum_{j=1}^{64} TP_j}{N}, \tag{19}$$

$$\sigma_{\overline{w}}^2 = \overline{w_{TP}}(1-\overline{w_{TP}}) = \overline{w_{TP}} - \overline{w_{TP}}^2, \tag{20}$$

где $\sigma_{\overline{w}}^2$ – дисперсия средней выборочной доли, TP_j – количество TP соответственно в j-й типической группе.

Известно следующее математическое неравенство Коши-Буняковского [32]:

$$\frac{x_1 + x_2 + ...x_n}{n} \le \sqrt{\frac{x_1^2 + x_2^2 + ...x_n^2}{n}}. \tag{21}$$

Если принять, что $x_1, x_2, ..., x_n > 0$, то возведем обе части неравенства (21) в квадрат и получим:

$$\left(\frac{\sum_{j=1}^{n} x_j}{n}\right)^2 \le \frac{\sum_{j=1}^{n} x_j^2}{n}. \tag{22}$$

Теперь, если принять в (22) $n = 64, x_j = w_{TP_j}$, а также воспользоваться формулами (18), (20), то получаем:

$$\overline{w_{TP}}^2 = \left(\frac{\sum_{j=1}^{64} w_{TP_j}}{64}\right)^2 \le \frac{\sum_{j=1}^{64} w_{TP_j}^2}{64} \Rightarrow \sigma_{\overline{w}}^2 \ge \overline{\sigma_w^2}. \tag{23}$$

Тем самым вычислив $\sigma_{\overline{w}}^2$, можно ограничить значение средней групповой дисперсии типических групп $\overline{\sigma_w^2}$.

Зная доверительные интервалы для $w_{TP}^*, w_{TN}^*, w_{FP}^*, w_{FN}^*$, можно также ограничить значения Re^*, Ac^*, Pr^* (истинные значения Re, Ac, Pr соответственно).

Оценка значения Re^*.

Согласно выражениям (4), (7) Re через w_{TP} выражается так:

$$Re = \frac{TP}{TP + FN} = \frac{TP}{N} = w_{TP}. \quad (24)$$

Тогда $Re^* = w_{TP}^*$.

Отсюда, используя неравенства (14) получаем:

$$Re - \Delta_{w_{TP}} \le Re^* \le Re + \Delta_{w_{TP}}. \quad (25)$$

Оценка значения Ac^*.

Согласно выражениям (6-8) Ac через w_{TP} и w_{TN} выражается так:

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{2N} = \frac{1}{2} w_{TP} + \frac{1}{2} w_{TN}. \quad (26)$$

Тогда $Ac^* = \frac{1}{2} w_{TP}^* + \frac{1}{2} w_{TN}^*$.

Согласно неравенствам (14):

$$Ac^* = \frac{1}{2} w_{TP}^* + \frac{1}{2} w_{TN}^* \le \frac{1}{2}(w_{TP} + \Delta_{w_{TP}}) + \frac{1}{2}(w_{TN} + \Delta_{w_{TN}}) =$$
$$= \frac{1}{2} w_{TP} + \frac{1}{2} w_{TN} + \frac{\Delta_{w_{TP}} + \Delta_{w_{TN}}}{2} = Ac + \frac{\Delta_{w_{TP}} + \Delta_{w_{TN}}}{2}.$$

Аналогично вычисляется с другой стороны.

Отсюда получается:

$$Ac - \frac{\Delta_{w_{TP}} + \Delta_{w_{TN}}}{2} \le Ac^* \le Ac + \frac{\Delta_{w_{TP}} + \Delta_{w_{TN}}}{2}. \quad (27)$$

В частном случае, если $\Delta_{w_{TP}} = \Delta_{w_{TN}} = \Delta$ выражение (27) принимает вид:

$$Ac - \Delta \le Ac^* \le Ac + \Delta .$$

Оценка значения Pr^*.

Согласно выражениям (5), (7-8), Pr через w_{FP} и w_{TN} выражается так:

$$Pr = \frac{TP}{TP + FP} = \frac{\frac{TP}{N}}{\frac{TP}{N} + \frac{FP}{N}} = \frac{w_{TP}}{w_{TP} + w_{FP}} = \frac{1}{1 + \frac{w_{FP}}{w_{TP}}}. \quad (28)$$

Тогда $Pr^* = \frac{1}{1 + \frac{w_{FP}^*}{w_{TP}^*}}$.

Согласно неравенствам (14), имеем:

$$Pr^* = \frac{1}{1 + \frac{w_{FP}^*}{w_{TP}^*}} \le \frac{1}{1 + \frac{w_{FP} - \Delta_{wFP}}{w_{TP} + \Delta_{wTP}}} = \frac{w_{TP} + \Delta_{wTP}}{w_{TP} + w_{FP} + \Delta_{wTP} - \Delta_{wFP}}. \quad (29)$$

$$Pr^* = \frac{1}{1 + \frac{w_{FP}^*}{w_{TP}^*}} \ge \frac{1}{1 + \frac{w_{FP} + \Delta_{wFP}}{w_{TP} - \Delta_{wTP}}} = \frac{w_{TP} - \Delta_{wTP}}{w_{TP} + w_{FP} + \Delta_{wFP} - \Delta_{wTP}}. \quad (30)$$

Неравенство (30) верно, при $w_{TP} > \Delta_{wTP}$.

Объединяя (29) и (30), получим:

$$\frac{w_{TP} - \Delta_{wTP}}{w_{TP} + w_{FP} + \Delta_{wFP} - \Delta_{wTP}} \le Pr^* \le \frac{w_{TP} + \Delta_{wTP}}{w_{TP} + w_{FP} + \Delta_{wTP} - \Delta_{wFP}}. \quad (31)$$

В частном случае, если ошибки $\Delta_{wFP} = \Delta_{wTP} = \Delta$ получим из (31) следующие выражения:

$$Pr^* \le \frac{w_{TP}+\Delta}{w_{TP}+w_{FP}} = \frac{w_{TP}}{w_{TP}+w_{FP}} + \frac{\Delta}{w_{TP}+w_{FP}} = Pr + \frac{\Delta}{w_{TP}+w_{FP}},$$

$$Pr^* \ge \frac{w_{TP}-\Delta}{w_{TP}+w_{FP}} = \frac{w_{TP}}{w_{TP}+w_{FP}} - \frac{\Delta}{w_{TP}+w_{FP}} = Pr - \frac{\Delta}{w_{TP}+w_{FP}}.$$

Как видно, ошибка Pr^* зависит от $w_{TP}+w_{FP}$. Эта зависимость проиллюстрирована в таблице 3.

Таблица 3. Зависимость ошибки Pr^* от $w_{TP}+w_{FP}$ при $\Delta_{wFP}=\Delta_{wTP}=\Delta$

$w_{TP}+w_{FP}$	Pr^*
≥ 1	$Pr-\Delta \le Pr^* \le Pr+\Delta$
≥ 0.5	$Pr-2\Delta \le Pr^* \le Pr+2\Delta$
≥ 0.1	$Pr-10\Delta \le Pr^* \le Pr+10\Delta$

Из таблицы 3 видно, что при малых значениях $w_{TP}+w_{FP}$ необходимо увеличивать выборку для снижения возможной ошибки.

Таким образом, исходя из того, какой по объему набор тестовых данных, можно с определённой долей вероятности найти доверительный интервал, в котором находятся истинные значения показателей действенности Re^*, Ac^*, Pr^*. Предельные ошибки выборки для каждого показателя $\Delta_e = \langle \Delta_{e1}, \Delta_{e2}.\Delta_{e3} \rangle$, исходя из (25), (27), (31) вычисляются следующим образом:

$$\Delta_{e1} = \Delta_{Re} = \Delta_{wTP}, \tag{32}$$

$$\Delta_{e2} = \Delta_{Pr} = max\left(\frac{w_{TP}+\Delta_{wTP}}{w_{TP}+w_{FP}+\Delta_{wTP}-\Delta_{wFP}} - Pr; Pr - \frac{w_{TP}-\Delta_{wTP}}{w_{TP}+w_{FP}+\Delta_{wFP}-\Delta_{wTP}} \right), \tag{33}$$

$$\Delta_{e3} = \Delta_{Ac} = \frac{\Delta_{wTP}+\Delta_{wTN}}{2}. \tag{34}$$

На основе рассуждений, представленных в разделах 4-6, сформируем этапы методики оценивания результативности функционирования СОВБ. Они представлены на рисунке 4.

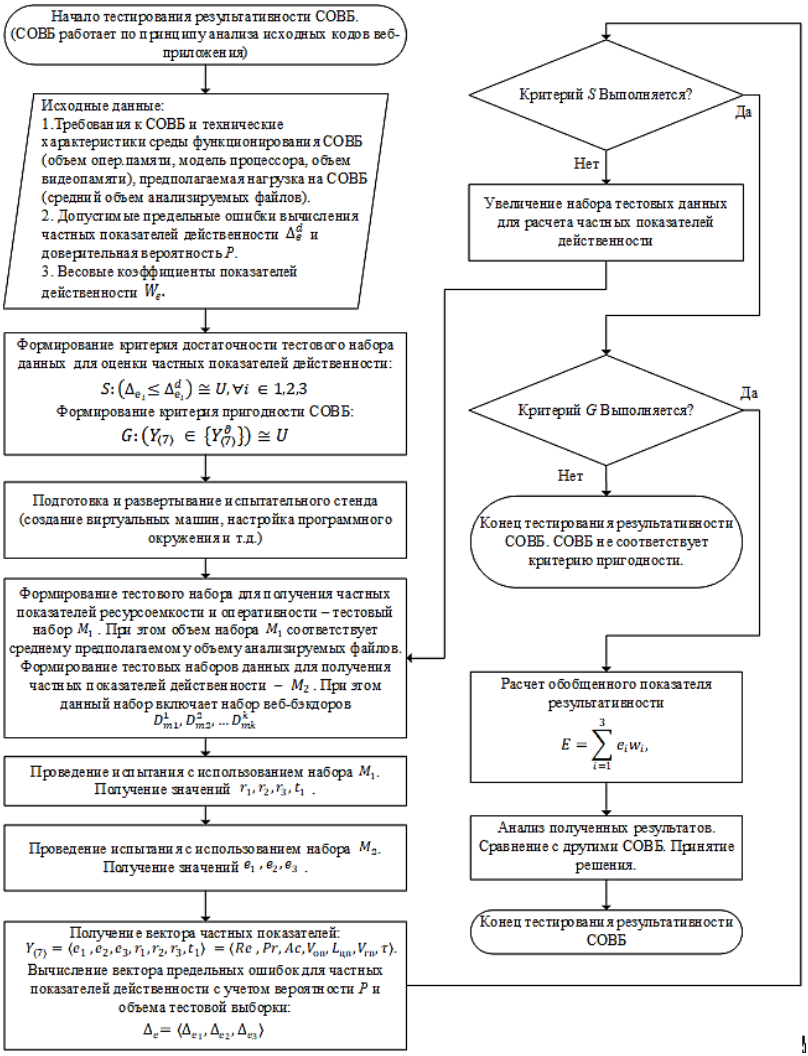

Рис. 4. Этапы методики оценивания результативности функционирования СОВБ

7. Проведение эксперимента и апробация методики. Перед тем, как полностью привести все шаги эксперимента, рассмотрим пример расчета предельных ошибок частных показателей $\Delta_e = \langle \Delta_{e1}, \Delta_{e2}, \Delta_{e3} \rangle$, исходя из объема тестовых наборов данных.

Пример вычисления $\Delta_e = \left\langle \Delta_{e_1}, \Delta_{e_2}, \Delta_{e_3} \right\rangle$.

В ходе тестирования алгоритма, представленного в разделе 3 настоящей статьи, использовался набор данных, состоящий из 128 объектов (64 легитимных файлов и 64 веб-бэкдоров). Данный алгоритм показал результаты, представленные в таблице 4.

Таблица 4. Результат проверки алгоритма обнаружения на тестовом наборе данных

		Прогноз наличия веб-бэкдора СОВБ	
		-	+
Фактическое наличие веб-бэкдора	-	62	2
	+	37	27

Исходя из формул (4-6) получаем частные показатели действенности: $e_1 = Re = 0,422$; $e_2 = Pr = 0,931$; $e_3 = Ac = 0,695$.

Вычислим предельные ошибки показателей. Коэффициент доверия z будем вычислять для $P = 0,997$.

Согласно (15) и (23):

$$\Delta_{w_{TP}} = \Delta_{w_{FN}} = z\sqrt{\frac{\overline{\sigma_w^2}}{64}} \le z\sqrt{\frac{\sigma_{\overline{w}}^2}{64}} = z\sqrt{\frac{\overline{w}\left(1-\overline{w}\right)}{64}} = 3\sqrt{\frac{\frac{27}{64}\cdot\frac{37}{64}}{64}} \approx 0,185.$$

Отсюда, согласно неравенствам (14):

$$0,422 - 0,185 \le w_{TP}^* \le 0,422 + 0,185,$$

$$0,578 - 0,185 \le w_{FN}^* \le 0,578 + 0,185.$$

Согласно (16) для w_{FP}^* и w_{TN}^*:

$$\Delta_{w_{FP}} = \Delta_{w_{TN}} = z\sqrt{\frac{\sigma_w^2}{64}} = z\sqrt{\frac{w(1-w)}{64}} = 3\sqrt{\frac{\frac{62}{64}\cdot\frac{2}{64}}{64}} \approx 0,065 .$$

Отсюда, согласно неравенствам (14):

$$0,031 - 0,065 \le w_{FP}^* \le 0,031 + 0,065 ,$$

$$0,969 - 0,065 \le w_{TN}^* \le 0,969 + 0,065 .$$

Отсюда, согласно (25), (27), (31):

$$0,422 - 0,185 \le Re^* \le 0,422 + 0,185 ,$$
$$0,711 \le Pr^* \le 1 ,$$
$$0,695 - 0,125 \le Ac^* \le 0,695 + 0,125 .$$

Тогда вектор ошибок для частных показателей действенности, согласно (32-34) будет выглядеть так: $\Delta_e = \langle 0,185; 0,220; 0,125 \rangle$.

Проведение эксперимента.

Необходимо вычислить обобщённые показатели результативности для трех СОВБ – *WebShellKiller, WEBSHELL.PUB, CloudWalker* [33], и выбрать наилучшее средство на основе сравнения этих показателей. Предъявлены следующие требования:

1) СОВБ должно работать в ОС *Ubuntu 20.04.5* с 8 Гб ОЗУ и 4-ядерным процессором *Intel core i7-10750.* Веб-приложение, которое анализирует СОВБ, основано на фреймворке WordPress. При этом:

– Среднее время анализа файла СОВБ не должно превышать 50 мс.

– Максимальный объем оперативной памяти, потребляемый средством, не должен превышать 500 Мб.

– Максимальный объем видеопамяти, потребляемый средством, не должен превышать 500 Мб.

– Максимальная загруженность процессора во время работы средства не должна превышать 30%.

2) Предельная ошибка для каждого вычисленного частного показателя действенности не должна превышать 0.05. Вероятность

нахождения частных показателей в пределах доверительного интервала равна $P = 0,997$.

3) Каждый из частных показателей действенности равнозначен (это равносильно тому, что весовые коэффициенты w_{e_i} для трех частных показателей равны 1/3).

Исходя из исходных требований, получаем, что область допустимых значений частных показателей:

$$\begin{aligned}\left\{Y^d_{\langle 7\rangle}\right\} = \left\{y^d_1 = \{0 \le Re \le 1\}, y^d_2 = \{0 \le Pr \le 1\}, y^d_3 = \{0 \le Ac \le 1\},\right.\\ \left. y^d_4 = \{V_{\text{оп}} \le 500\text{Мб}\}, y^d_5 = \{L_{\text{цп}} \le 30\%\}, y^d_6 = \{L_{\text{гп}} \le 500\text{Мб}\}, y^d_7 = \{\tau \le 50\text{мс}\}\right\}.\end{aligned} \quad (35)$$

Вектор допустимых предельных ошибок частных показателей действенности:

$$\Delta^d_e = \left\langle \Delta^d_{e1}, \Delta^d_{e2}, \Delta^d_{e3} \right\rangle = \left\langle 0,05; 0,05; 0,05 \right\rangle. \quad (36)$$

Вектор весовых коэффициентов для частных показателей действенности, в виду их равнозначности:

$$W_e = \left\langle w_{e1}, w_{e2}, w_{e3} \right\rangle = \left\langle \frac{1}{3}; \frac{1}{3}; \frac{1}{3} \right\rangle. \quad (37)$$

Тестирование СОВБ проводится следующим образом:

1 этап. Вычисление показателей e_1, e_2, e_3 на наборе легитимных файлов и веб-бэкдоров (с учетом выборки, представленной в подразделе 6.1).

2 этап. Вычисление показателей r_1, r_2, r_3 и t_1 на всем объеме файлов фреймворка WordPress.

Для первого этапа используется 96 файлов, содержащих веб-бэкдоры, которые были перечислены ранее [28] (96 файлов потому, что 32 веб-бэкдора состоят из одного файла и 32 веб-бэкдора – из двух файлов), а также 64 файлов, которые не содержат веб-бэкдоров. В качестве незараженных файлов использовались файлы из того же фреймворка *WordPress*. В процессе расчета частных показателей действенности веб-бэкдоры, которые состоят из двух файлов, рассматриваются как единый объект. Таким образом, для обнаружения такого веб-бэкдора достаточно определить его наличие хотя бы

в одном из двух файлов. В итоге получилось 128 объектов для тестирования СОВБ.

СОВБ проводит анализ файлов и выдает результат о наличии или отсутствии веб-бэкдора в каждом из них. После чего можно вычислить показатели $e_1 = Re, e_2 = Pr, e_3 = Ac$ и предельные ошибки выборки $\Delta_e = \langle \Delta_{e1}, \Delta_{e2}.\Delta_{e3} \rangle$.

Для второго этапа использовался набор из всех файлов PHP, из которых состоит сайт на фреймворке *WordPress*, с добавлением также всех созданных веб-бэкдоров. В конечном итоге всего получилось 1191 файл.

Испытательный стенд представляет собой виртуальную машину *Ubuntu 20.04.5* с 8 Гб ОЗУ и 4-ядерным процессором *Intel core i7-10750H*.

На первом этапе были получены TP, TN, FP, FN. С помощью них были вычислены значения частных показателей действенности Re, Pr, Ac с помощью формул (4-6). Также были вычислены доверительные интервалы каждого из показателей. Это можно увидеть в таблице 5. Предельные ошибки частных показателей действенности, исходя из объема выборки, равны:

$$\begin{aligned} \Delta_e^1 &= \langle 0{,}146; 0{,}516; 0{,}124 \rangle, \\ \Delta_e^2 &= \langle 0{,}184; 0{,}276; 0.138 \rangle, \\ \Delta_e^3 &= \langle 0{,}124; 0{,}566; 0{,}117 \rangle, \end{aligned} \tag{38}$$

где 1 – *WebShellKiller,* 2 – *WEBSHELL.PUB,* 3 – *CloudWalker.*

На втором этапе каждая из СОВБ получала на вход набор данных из 1191 файла. Полученные значения показателей ресурсоемкости и оперативности также представлены в таблице 5. (Показатель $V_{\text{гп}}$ для каждой системы равен 0, потому что ни одна из систем не использует видеопамять во время работы.)

На основании измеренных значений получаем вектора частных показателей для СОВБ:

$$\begin{aligned} Y^1 &= \langle 0{,}188; 0{,}706; 0{,}555; 35\text{Мб}; 20\%; 0\text{Мб}; 3.8\text{мс} \rangle, \\ Y^2 &= \langle 0{,}406; 0{,}867; 0{,}672; 58\text{Мб}; 24\%; 0\text{Мб}; 1.9\text{мс} \rangle, \\ Y^3 &= \langle 0{,}125; 0{,}571; 0{,}516; 156\text{Мб}; 39\%; 0\text{Мб}; 40{,}4\text{мс} \rangle. \end{aligned} \tag{39}$$

Таблица 5. Результаты тестирования СОВБ

Показатели	**WebShellKiller**	**WEBSHELL.PUB**	**CloudWalker**
TP	12	26	8
FP	5	4	6
TN	59	60	58
FN	52	38	56
Частные показатели действенности с доверительными интервалами			
$e_1(Re)$	0,188; $0,042 \le Re^* \le 0,334$	0,406; $0,222 \le Re^* \le 0,590$	0,125; $0,001 \le Re^* \le 0,249$
$e_2(Pr)$	0,706; $0,190 \le Pr^* \le 1$	0,867; $0,591 \le Pr^* \le 1$	0,571; $0,005 \le Pr^* \le 1$
$e_3(Ac)$	0,555; $0,431 \le Ac^* \le 0,679$	0,672; $0,534 \le Ac^* \le 0,810$	0,516; $0,399 \le Ac^* \le 0,633$
Частные показатели ресурсоемкости			
$r_1(V_{\text{оп}})$	35 Мб	58 Мб	156 Мб
$r_2(V_{\text{цп}})$	20%	24%	39%
$r_3(V_{\text{гп}})$	0 Мб	0 Мб	0 Мб
Частные показатели оперативности			
$t_1(\tau)$	3,8 мс	1,9 мс	40,4 мс

Однако полученные значения не соответствуют критерию достаточности тестовых наборов веб-бэкдоров (2). Каждая предельная ошибка частного показателя действенности (38) больше, чем 0.05. Тем самым объем тестовых данных для вычисления показателей частных показателей действенности необходимо увеличить. Затем проводить эксперимент нужно заново. Однако в целях демонстрации продолжим вычисления, предполагая, что критерий достаточности тестовых наборов все-таки выполнен.

Примечание

Как можно видеть, возможные ошибки частных показателей действенности (38) получились достаточно большими, потому что в качестве набора данных использовался всего один пакет – 128 объектов. Увеличение объема тестовых данных (легитимных файлов и веб-бэкдоров), согласно выборке, предложенной в подразделе 6.1, приведет к сужению диапазона доверительных интервалов, и тем самым к уменьшению предельных ошибок частных показателей.

Исходя из области допустимых значений (35) очевидно, что третья СОВБ *CloudWalker* исключается из эксперимента, так как $L_{\text{цп}} = 39\% > 30\%$.

На основе вектора весовых коэффициентов (37) и формулы (1) вычислим обобщенные значения показателей результативности для первого и второго СОВБ:

$$E^1 = \frac{1}{3}0,188 + \frac{1}{3}0,706 + \frac{1}{3}0,555 \approx 0,483,$$
$$E^2 = \frac{1}{3}0,406 + \frac{1}{3}0,867 + \frac{1}{3}0,672 \approx 0,648. \quad (40)$$

Как видно из (40) второе средство обладает более высоким значением результативности. Поскольку частные показатели действенности находятся в одном диапазоне значений от 0 до 1, в идеальном СОВБ обобщенный показатель результативности будет иметь значение 1. Это достигается, когда все частные показатели действенности равны 1. Таким образом, для заданных условий WEBSHELL.PUB лучше всего подходит в качестве СОВБ, однако его интегральная результативность не слишком высока (0.648, при максимальном значении равном 1).

8. Заключение. Многие исследователи производят оценку СОВБ только на основе собственных наборов данных, что делает эту оценку не полностью объективной, что было показано в работе [4]. В настоящей статье предложена методика, позволяющая производить объективное оценивание результативности функционирования СОВБ. В методике выделены три группы частных показателей, используемых для оценки СОВБ: действенность, ресурсоемкость (затраты ресурсов) и оперативность (затраты времени) функционирования. Частные показатели ресурсоемкости и оперативности непосредственно не участвуют в вычислении обобщенного показателя результативности, однако они используются при определении пригодности СОВБ. Частные показатели действенности используются для расчета показателя результативности СОВБ.

Для формирования тестовых данных была разработана классификация веб-бэкдоров, встроенных в исходный код веб-приложений. На основе объема тестового набора данных, полученного с помощью специальной выборки, представленной в разделе 6, вычисляются доверительные интервалы частных показателей действенности и соответствующие предельные ошибки этих значений. Таким образом, в методике также предусмотрен критерий достаточности тестовых наборов данных. Объективность оценивания результативности функционирования СОВБ заключается в том, что для создания набора тестовых данных применяется обобщенная классификация веб-бэкдоров, встроенных в исходный код веб-приложений, а также рассчитываются доверительные интервалы для значений частных показателей действенности.

Разработанная методика применима для СОВБ, которые работают на основе анализа исходного кода веб-страниц. Для ее использования необходимы определенные исходные данные, такие как допустимые предельные ошибки частных показателей действенности и вероятность их нахождения в доверительном интервале, а также весовые коэффициенты частных показателей действенности, которые определяются экспертными методами. В результате применения методики вычисляется обобщенный показатель результативности, который зависит от весовых коэффициентов частных показателей. Если частные показатели равнозначны, весовые коэффициенты принимают значение 1/3. Обобщенный показатель результативности варьируется от 0 до 1, при этом максимальное значение «1» указывает на то, что данное СОВБ при заданных условиях является максимально результативным и способно обнаружить любой веб-бэкдор, встроенный в исходный код веб-приложения.

Литература

1. Актуальные киберугрозы: итоги 2022 года. URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2022/ (дата обращения: 02.12.2023).
2. Albalawi M.M., Aloufi R.B., Alamrani N.A., Albalawi N.N., Aljaedi O.A., Alharbi A.R. Website Defacement Detection and Monitoring Methods: A Review // Electronics. 2022. vol. 11(21). DOI: /10.3390/electronics11213573.
3. Кибербезопасность в 2023–2024 гг.: тренды и прогнозы. Часть третья. URL: https://www.ptsecurity.com/ru-ru/research/analytics/kiberbezopasnost-v-2023-2024-gg-trendy-i-prognozy-chast-tretya/#id2 (дата обращения: 02.02.2024).
4. Боровков В.Е., Ключарёв П.Г. Методы защиты веб-приложений от злоумышленников // Вопросы кибербезопасности. 2023. № 5(57). С. 89–99.
5. ГОСТ Р ИСО 9000-2015. Системы менеджмента качества. Основные положения и словарь // М.: Стандартинформ. 2018.
6. Sam L.T., Aurelien F. Backdoors: Definition, Deniability and Detection // Research in Attacks, Intrusions, and Defenses (RAID 2018). 2018. pp. 92–113.
7. Киселев А.Н. Подход к обнаружению вредоносного программного обеспечения web-shell на основе анализа сетевого трафика web-инфраструктуры // Труды Военно-космической академии имени А.Ф. Можайского. 2021. № 677. С. 143–152.
8. Ma M., Han L., Zhou C. Research and application of artificial intelligence based webshell detection model: A literature review // arXiv preprint arXiv.2405.00066. 2024.
9. Omer A. Performance Comparison of Static Malware Analysis Tools Versus Antivirus Scanners To Detect Malware // 1 Performance Comparison of Static Malware Analysis Tools Versus Antivirus Scanners To Detect Malware. 2017. pp. 1–6.
10. Real-World Protection Test July-October 2022. URL: https://www.av-comparatives.org/tests/real-world-protection-test-july-october-2022/ (дата обращения: 02.12.2023)
11. Лысенко А.В., Кожевникова И.С., Ананьин Е.В. Анализ методов обнаружения вредоносных программ // Молодой ученый. 2016. № 21(125). С. 758–761.

12. Fu J, Li L., Wang Y. Webshell Detection Based on Convolutional Neural Network // Journal of Zhengzhou University (Natural Science Edition). 2019. vol. 51(2). pp. 1–8.
13. WebShell detection based on semantic features. URL: https://litheory.github.io/publication/webshell-detection-based-on-semantic-features/ (дата обращения: 11.01.2024).
14. Word2vec. URL: https://www.tensorflow. org/text/tutorials/word2vec (дата обращения: 11.01.2024).
15. Pan Z., Chen Y., Chen Y., Shen Y., Guo X. Webshell Detection Based on Executable Data Characteristics of PHP Code // Wireless Communications and Mobile Computing. 2021. no. 1. pp. 1–12. DOI: 10.1155/2021/5533963.
16. Kaushik K., Aggarwal S., Mudgal S., Saravgi S., Mathur V. A novel approach to generate a reverse shell: Exploitation and Prevention // International Journal of Intelligent Communication, Computing, and Networks. 2021. vol. 2. DOI: 10.51735/ijiccn/001/33.
17. p0wnyshell – Single-file PHP Shell. URL: https://github.com/flozz/p0wny-shell (дата обращения: 12.01.2024).
18. WordPress. URL: http://wordpress.com (дата обращения: 12.01.2024).
19. Obfuscation Techniques in MARIJUANA Shell «Bypass». URL: https://blog.sucuri.net/2020/12/obfuscation-techniques-in-marijuana-shell-bypass.html (дата обращения: 20.01.2024).
20. Keeping Web Shells under Cover (Web Shells Part 3). URL: https://www.acunetix.com/blog/articles/keeping-web-shells-undercover-an-introduction-to-web-shells-part-3/ (дата обращения: 21.01.2024).
21. Петухов Г.Б., Якунин В.И. Методологические основы внешнего проектирования целенаправленных процессов и целеустремленных систем. М.: АСТ. 2006. 504 с.
22. Гаценко О.Ю., Мирзабаев А.Н., Самонов А.В. Методы и средства оценивания качества реализации функциональных и эксплуатационно-технических характеристик систем обнаружения и предупреждения вторжений нового поколения // Вопросы кибербезопасности. 2018. № 2(26). С. 24–32.
23. Zhu T., Weng Z., Fu L., Ruan L. A Web Shell Detection Method Based on Multiview Feature Fusion // Applied Sciences. 2020. vol. 10(18). DOI: 10.3390/app10186274.
24. Pu A., Feng X.., Zhang Y., Wan X., Han J., Huang C. BERT-Embedding-Based JSP Webshell Detection on Bytecode Level Using XGBoost // Security and Communication Networks. 2022. pp. 1–11. DOI: 10.1155/2022/4315829.
25. Nguyen N., Le V., Phung V., Du P. Toward a Deep Learning Approach for Detecting PHP Webshell // SoICT 2019: Proceedings of the Tenth International Symposium on Information and Communication Technology. 2019. pp. 514–521. DOI: 10.1145/3368926.3369733.
26. Zhang H., Liu M., Yue Z., Xue Z., Shi Y., He X. A PHP and JSP Web Shell Detection System with Text Processing Based on Machine Learning // 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). 2020. pp. 1584–1591.
27. Оценка качества в задачах классификации и регрессии. URL: https://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задачах_классификации_и_регрессии (дата обращения: 15.11.2023).
28. DS_WBDT. URL: https://github.com/scienceMGtech (дата обращения: 20.10.2023).
29. Zhao J., Lu J., Wang X., Zhu K., Yu L. WTA: A Static Taint Analysis Framework for PHP Webshell // Applied Sciences. 2021. vol. 11(16). DOI: 10.3390/app11167763.
30. Ниворожкина Л.И. и др. Статистические методы анализа данных // РИОР, 2016. 333 с.

31. Илышев А.М. Общая теория статистики. Учебник для студентов вузов, обучающихся по специальностям экономики и управления. М.: ЮНИТИ. 2012. 535 с.
32. Соловьев Ю.П. Неравенства. М.: МЦНМО, 2005. 16 с.
33. WebShell Scan Detection and Killing Tools. URL: https://cloud.tencent.com/developer/article/1745883 (дата обращения: 27.02.2024).

Боровков Владислав Евгеньевич — аспирант кафедры, кафедра «информационной безопасности», Московский Государственный Технический Университет имени Н.Э. Баумана. Область научных интересов: машинное обучение, глубокое обучение, информационная безопасность, оценка безопасности компьютерных систем. Число научных публикаций — 14. vbscience@yandex.ru; 2-я Бауманская улица, 5/4, 105005, Москва, Россия; р.т.: +7(499)263-6936.

Ключарёв Петр Георгиевич — д-р техн. наук, профессор кафедры, кафедра «информационной безопасности», Московский Государственный Технический Университет имени Н.Э. Баумана. Область научных интересов: криптография, теоретическая информатика, дискретная математика, информационная безопасность. Число научных публикаций — 75+. pk.iu8@yandex.ru; 2-я Бауманская улица, 5/4, 105005, Москва, Россия; р.т.: +7(499)263-6936.

Денисенко Денис Игоревич — независимый исследователь информационной безопасности, разработчик программного обеспечения. Область научных интересов: информационная безопасность, анализ защищённости информационных систем, высоконагруженные приложения, безопасность приложений. Число научных публикаций — 2. researchDD_journal@mail.ru; 2-я Бауманская улица, 5/4, 105005, Москва, Россия; р.т.: +7(499)263-6936.

DOI 10.15622/ia.24.1.6

V. BOROVKOV, P. KLYUCHAREV, D. DENISENKO

TECHNIQUE FOR ASSESSING THE EFFECTIVENESS OF THE FUNCTIONING OF WEB BACKDOOR DETECTION SYSTEMS

Borovkov V., Klyucharev P., Denisenko D. **Technique for Assessing the Effectiveness of the Functioning of Web Backdoor Detection Systems.**

Abstract. Currently, there is a significant increase in information security incidents related to attacks on web resources. Obtaining unauthorized access to web resources remains one of the main methods of penetration into corporate networks of organizations and expanding the capabilities of intruders. In this regard, many studies are aimed at developing web backdoor detection systems (WBDS), but there is a need to assess the effectiveness of these systems. The purpose of this study is to develop an objective approach to assess the effectiveness of the WBDS functioning. In this work, it was found that the effectiveness of web backdoor detection systems is objectively manifested in the process of their use, therefore, testing of such systems should be carried out in conditions as close as possible to real ones. In this regard, the article proposes a new technique for assessing the effectiveness of WBDS. It is based on the calculation of three groups of specific indicators characterizing the potency, resource intensity and responsiveness of the detection tool, as well as the calculation of a generalized effectiveness indicator. Based on an analysis of research in this area, a classification of web backdoors embedded by an attacker into the source code of web applications has been developed. This classification is used when generating test datasets to calculate specific potency indicators. The developed methodology is applicable to tools that work based on the analysis of the source code of web pages. Additionally, its use requires a number of initial data, such as permissible maximum errors of frequent potency indicators and the probability of them being within the confidence interval, as well as weighting coefficients of specific potency indicators, which are selected by expert methods. This work may be useful for information security specialists and researchers who want to conduct a more objective assessment of their WBDS.

Keywords: cybersecurity, web vulnerabilities, web backdoors, web shells, machine learning, testing methods and tools.

References

1. Aktual'nye kiberugrozy: itogi 2022 goda [Current cyber threats: the results of 2022]. Available at: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2022/ (accessed: 02.12.2023). (In Russ.).
2. Albalawi M.M., Aloufi R.B., Alamrani N.A., Albalawi N.N., Aljaedi O.A., Alharbi A.R. Website Defacement Detection and Monitoring Methods: A Review. Electronics. 2022. vol. 11(21). DOI: 10.3390/electronics11213573.
3. Kiberbezopasnost v 2023–2024 gg.: trendy i prognozy. Chast tretia [Cybersecurity in 2023-2024: trends and forecasts. Part Three]. Available at: https://www.ptsecurity.com/ru-ru/research/analytics/kiberbezopasnost-v-2023-2024-gg-trendy-i-prognozy-chast-tretya/#id2 (accessed: 02.02.2024). (In Russ.).
4. Borovkov V., Klyucharev P. [Methods of protecting web applications from intruders]. Voprosy kiberbezopasnosti – Issues of cybersecurity. 2023. no. 5(57). pp. 89–99. (In Russ.).
5. GOST R ISO 9000-2015. [Quality management systems. Basic provisions and vocabulary] // M.: Standartinform. 2018. (In Russ.).

6. Sam L.T., Aurelien F. Backdoors: Definition, Deniability and Detection. Research in Attacks, Intrusions, and Defenses (RAID 2018). 2018. pp. 92–113.
7. Kiselev A.N. [An approach to the detection of malicious web-shell software based on the analysis of network traffic of the web infrastructure]. Trudy Voenno-kosmicheskoj akademii imeni A.F. Mozhajskogo – Proceedings of the Military Space Academy named after A.F. Mozhaisky. 2021. no. 677. pp. 143–152. (In Russ.).
8. Ma M., Han L., Zhou C. Research and application of artificial intelligence based webshell detection model: A literature review. arXiv preprint arXiv.2405.00066. 2024.
9. Omer A. Performance Comparison of Static Malware Analysis Tools Versus Antivirus Scanners To Detect Malware. 1 Performance Comparison of Static Malware Analysis Tools Versus Antivirus Scanners To Detect Malware. 2017. pp. 1–6.
10. Real-World Protection Test July-October 2022. Available at: https://www.av-comparatives.org/tests/real-world-protection-test-july-october-2022/ (accessed: 02.12.2023)
11. Lysenko A., Kozhevnikova I., Ananyin E. [Analysis of malware detection methods]. Molodoj uchenyj – Young Scientist. 2016. no. 21(125). pp. 758–761. (In Russ.).
12. Fu J, Li L., Wang Y. Webshell Detection Based on Convolutional Neural Network. Journal of Zhengzhou University (Natural Science Edition). 2019. vol. 51(2). pp. 1–8.
13. WebShell detection based on semantic features. Available at: https://litheory.github.io/publication/webshell-detection-based-on-semantic-features/ (accessed: 11.01.2024).
14. Word2vec. Available at: https://www.tensorflow. org/text/tutorials/word2vec (accessed: 11.01.2024).
15. Pan Z., Chen Y., Chen Y., Shen Y., Guo X. Webshell Detection Based on Executable Data Characteristics of PHP Code. Wireless Communications and Mobile Computing. 2021. no. 1. pp. 1–12. DOI: 10.1155/2021/5533963.
16. Kaushik K., Aggarwal S., Mudgal S., Saravgi S., Mathur V. A novel approach to generate a reverse shell: Exploitation and Prevention. International Journal of Intelligent Communication, Computing, and Networks. 2021. vol. 2. DOI: 10.51735/ijiccn/001/33.
17. p0wnyshell – Single-file PHP Shell. Available at: https://github.com/flozz/p0wny-shell (accessed: 12.01.2024).
18. WordPress. Available at: http://wordpress.com (accessed: 12.01.2024).
19. Obfuscation Techniques in MARIJUANA Shell «Bypass». Available at: https://blog.sucuri.net/2020/12/obfuscation-techniques-in-marijuana-shell-bypass.html (accessed: 20.01.2024).
20. Keeping Web Shells under Cover (Web Shells Part 3). Available at: https://www.acunetix.com/blog/articles/keeping-web-shells-undercover-an-introduction-to-web-shells-part-3/ (accessed: 21.01.2024).
21. Petukhov G., Yakunin V. Metodologicheskie osnovy vneshnego proektirovanija celenapravlennyh processov i celeustremlennyh sistem [Methodological foundations of external design of purposeful processes and purposeful systems]. Moscow: AST. 2006. 504 p. (In Russ.).
22. Gatsenko O., Mirzabaev A., Samsonov A. [Methods and tools for evaluating the quality of implementation of functional and operational and technical characteristics of intrusion detection and prevention systems of a new generation]. Voprosy kiberbezopasnosti – Cybersecurity issues. 2018. no. 2(26). pp. 24–32. (In Russ.).
23. Zhu T., Weng Z., Fu L., Ruan L. A Web Shell Detection Method Based on Multiview Feature Fusion. Applied Sciences. 2020. vol. 10(18). DOI: 10.3390/app10186274.

24. Pu A., Feng X.., Zhang Y., Wan X., Han J., Huang C. BERT-Embedding-Based JSP Webshell Detection on Bytecode Level Using XGBoost. Security and Communication Networks. 2022. pp. 1–11. DOI: 10.1155/2022/4315829.
25. Nguyen N., Le V., Phung V., Du P. Toward a Deep Learning Approach for Detecting PHP Webshell. SoICT 2019: Proceedings of the Tenth International Symposium on Information and Communication Technology. 2019. pp. 514–521. DOI: 10.1145/3368926.3369733.
26. Zhang H., Liu M., Yue Z., Xue Z., Shi Y., He X. A PHP and JSP Web Shell Detection System with Text Processing Based on Machine Learning. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). 2020. pp. 1584–1591.
27. Ocenka kachestva v zadachax klassifikacii i regressii [Quality assessment in classification and regression tasks]. Available at: https://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задачах_классификации_и_регрессии (accessed: 15.11.2023). (In Russ.).
28. DS_WBDT. Available at: https://github.com/scienceMGtech (accessed: 20.10.2023).
29. Zhao J., Lu J., Wang X., Zhu K., Yu L. WTA: A Static Taint Analysis Framework for PHP Webshell. Applied Sciences. 2021. vol. 11(16). DOI: 10.3390/app11167763.
30. Nivorozhkina L.I. et al. Statisticheskiye metody analiza dannykh [Statistical methods of data analysis]. Moscow: RIOR. 2016. 333 p. (In Russ.).
31. Ilyshev A.M. Obshhaja teorija statistiki. Uchebnik dlja studentov vuzov, obuchajushhihsja po special'nostjam jekonomiki i upravlenija [General theory of statistics. Textbook for university students studying economics and management]. Moscow: UNITY. 2012. 535 p. (In Russ.).
32. Solovyev Yu. Neravenstva [Inequalities]. Moscow: MTsNMO. 2005. 16 p. (In Russ.).
33. WebShell Scan Detection and Killing Tools. Available at: https://cloud.tencent.com/developer/article/1745883 (accessed: 27.02.2024).

Borovkov Vladislav — Postgraduate student, Department of information security, Bauman Moscow State Technical University. Research interests: machine learning, deep learning, information security, computer system security assessment. The number of publications — 14. vbscience@yandex.ru; 5/4, 2nd Baumanskaya St., 105005, Moscow, Russia; office phone: +7(499)263-6936.

Klyucharev Peter — Ph.D., Dr.Sci., Professor of the department, Department of information security, Bauman Moscow State Technical University. Research interests: cryptography, theoretical computer science, discrete mathematics, information security. The number of publications — 75+. pk.iu8@yandex.ru; 5/4, 2nd Baumanskaya St., 105005, Moscow, Russia; office phone: +7(499)263-6936.

Denisenko Denis — Independent information security researcher, software developer. Research interests: information security, security analysis of information systems, high-load applications, application security. The number of publications — 2. researchDD_journal@mail.ru; 5/4, 2nd Baumanskaya St., 105005, Moscow, Russia; office phone: +7(499)263-6936.