электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 519.816:004.056.53

Анализ и оценка безопасности веб-приложения с применением операторов агрегирования

Павлов Н.С.*студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»

Научный руководитель: Сакулин С.А., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»
sakulin@bmstu.ru

Любые приложения, в том числе и веб, могут довольно сильно отличаться друг от друга, как в функциональном плане, так и в мерах безопасности. Это характеризуется разными задачами, которые на них возлагают. При разработке приходится учитывать множество факторов. Особую сложность представляют те факторы, которые заведомо неизвестны или могут менять свои характеристики в жизненном цикле программного обеспечения (ПО). Поэтому в данной статье используется агрегирование критериев, основанное на интеграле Шоке. Такое агрегирование может учитывать возможные неопределенности в данных.

Для определения областей безопасности, а также определения самих критериев использовались наработки открытого проекта обеспечения безопасности веб-приложений OWASP (Open Web Application Security Project). В сообщество OWASP входят образовательные организации, корпорации и частные лица со всего мира. В сферу деятельности сообщества входит создание статей, учебных пособий, документации, инструментов и технологий, находящихся в свободном доступе.

Одним из созданных стандартов является стандарт верификации безопасности приложения ASVS (Application Security Verification Standard) [1].

Основная цель ASVS - стандартизация диапазона охвата и уровня строгости доступных на рынке приложений, обеспечивающих безопасность.

Сообщество OWASP предлагает следующее использование стандартов ASVS:

1) Помощь организациям в разработке и поддержке безопасных приложений.
2) Установка требований безопасности между поставщиками и клиентами.

В данном стандарте критерии разбиты на группы по их общей направленности (например, идентификация, контроль доступа и т.д.). В стандарте указаны следующие **группы критериев:**

- Аутентификация (G_1);
- Управление сеансами (G_2);
- Контроль доступа (G_3);
- Вредоносная обработка ввода (G_4);
- Слабая криптография (G_5);
- Обработка ошибок и запись их в журнал (G_6);
- Защита данных (G_7);
- Безопасность связи (G_8);
- Безопасность HTTP (G_9);
- Вредоносные управления (G_{10});
- Бизнес-логика (G_{11});
- Файлы и ресурсы (G_{12});
- Мобильная (G_{13});

Каждая группа включает в себя критерии своей области (например, аутентификация включает контроль сложности пароля, проверка подлинности, контроль учётной записи и т.д.).

В данной статье не рассматриваются все критерии подробно т.к. их большое количество и решение об их применении ложится на разработчиков в зависимости от поставленных задач. Применим следующие обозначения для критериев: g_1,g_2, g_3 и т.д.. В качестве примера приведены несколько критериев из группы G_1 (Аутентификация):

- Требуется аутентификация везде, кроме страниц специально предназначенных для публичного доступа (g_1^1);
- Не выводится пароль, когда пользователь вошел в систему (g_2^1);
- Все проверки подлинности проводятся на стороне сервера ($g_{3)}^1$;
- Все элементы управления аутентификацией (в том числе библиотеки, которые требуют внешних служб аутентификации) имеют централизованную реализацию ($g_{4)}^1$;

– Средства управления идентификацией при сбое не позволят нападавшим авторизоваться (g_5^1);
– В области ввода позволяется или поощряется длинные или очень сложные пароли и обеспечивается достаточный минимум, чтобы защитить от применения часто используемых.

В группе G_1 (Аутентификация) присутствуют еще 15 критериев. Для работы с другими группами применим аналогичные обозначения (например, g_1^3 - для первого критерия 3-ей группы).

Рассматриваемые критерии можно оценить по-разному, например, рассматривать их как булевы переменные (приложение проходит по требованию, то оценка «да», в противном случае оценка «нет»), но это не позволит оценить приложение в целом. Поэтому в данной статье при оценке учитываются уровни из стандарта ASVS [1].

Изначально в ASVS описаны четыре уровня верификации требований безопасности приложения. Чтобы показать их, воспользуемся рисунком из стандарта ASVS (Рис.1). Уровни располагаются в порядке возрастания с 0-го до 3-го. Переход на каждый новый уровень означает, что все требования нижестоящих уровней выполнены.

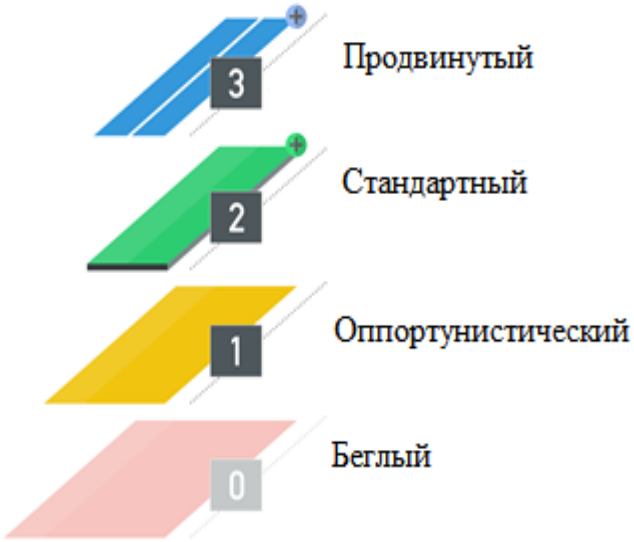

Рис. 1. Уровни верификации требований безопасности стандарта ASVS.

Рассмотрим уровни ASVS более подробно.

0-ой уровень - Беглый. Является необязательным для верификации, обозначает начальные требования безопасности, которые ставятся перед приложением (например, улучшенный механизм аутентификации). Поэтому этот уровень не учитывается ни в стандарте [1], ни в данной статье.

1-ый уровень – Оппортунистический. Приложение удовлетворяет первому уровню, если адекватно защищает от угроз безопасности, которые легко обнаружить. Применяется чаще всего для приложений, в которых злоумышленник не будет целенаправленно искать уязвимости или потеря данных не несет серьезного риска.

2-ой уровень – Стандарт. Этот уровень подходит для большинства приложений. В него входят: критерии уязвимости бизнес-логики, он также включает в себя список OWASP Top 10 – 2013 The Ten Most Critical Application Security Risks [2]. Он подразумевает, что злоумышленник будет целенаправленно искать уязвимости, используя специальные инструменты сканирования, а также методы ручного тестирования.

3-ий уровень – Продвинутый. Приложение удовлетворяет третьему уровню, если оно адекватно защищает от всех известных угроз безопасности приложений, а также демонстрирует принципы хорошего дизайна безопасности. Уровень 3, как правило, подходит для критически важных приложений, которые ответственны за жизнь человека или безопасность важных инфраструктур.

Приложение также может достигать уровня 3+. Если проверять только код веб-приложения, то речь идет о стандартных уровнях ASVS. Если же проверить и сторонние библиотеки или расширения, то можно говорить о полной защищенности приложения и ему дается категория уровня 3+.

Для универсальной модели требуется непредвзятая оценка. Для этого воспользуемся уровнями безопасности из стандарта ASVS [1]. В этом стандарте упоминается, что каждый уровень зависит от предыдущих в равной степени (третий уровень не достижим без выполнения всех требований 2-го и 1-го, второй не достижим без выполнения всех требований 1-го). Это говорит об одинаковом влиянии их между собой на конечную оценку. Следовательно, при оценке той или иной группы критериев и имея, например, шкалу от 0 до 10-ти, можно определить на ней границы уровней, поделив ее на 3. Поэтому для текущей шкалы от 0 до 10 будут следующие оценки:

- 3.33 – удовлетворение любого критерия 1-го уровня.
- 6.66 – выполнение критерия 2-го уровня и всех 1-го. 3.33 – выполнение критерия 2-го уровня и не выполнение всех критериев 1-го.

-9.99 – выполнение критерия 3-го уровня при выполнении всех 1-го и 2-го. В противном случае оценка 3.33 за удовлетворение критерия на 3-ем уровне.

Имея данный алгоритм оценки перейдем к операторам агрегирования. Оператором агрегирования часто называют обладающую некоторыми заданными свойствами функцию от H критериев, каждый из которых определен на единичном интервале. Областью

значений этой функции также является единичный интервал. В соответствии с этим, оператор агрегирования обозначают $AGG:[0,1]^H \to [0,1]$, где H - число критериев [4]. Таким образом, для формулирования оценок необходимо найти подходящий оператор $AGG(g_1^1,...,g_3^1,g_1^2,...g_2^2,g_1^3,...g_2^3,...g_1^{13},...,g_3^{13})$. Результатом агрегирования с помощью этого оператора будет оценка Ω безопасности веб-приложения в интервале [0,1].

Для работы с критериями на единичной шкале необходима нормализация всех критериев. Например, если оценка степени соответствия веб-приложения тому или иному критерию выставляется в соответствии с десятибалльной шкалой {0,1,…,9,10}, то нормализация этой шкалы даст в результате шкалу {0,0.1,…,0.9,1} и будет состоять в делении каждой возможной оценки на 10. В случае, если для разных критериев оценки используются разные шкалы, все эти шкалы необходимо свести к единичному отрезку (нормализовать).

Самым простым и наиболее распространенным на практике оператором агрегирования критериев является средневзвешенный оператор. Однако, этот оператор не позволяет формализовать знания о зависимостях между критериями [4]. Наиболее подходящим с практической точки зрения представляется применение интеграла Шоке 2-го порядка, поскольку он позволяет формализовать рассуждения эксперта, оставаясь при этом достаточно простым. В [5] рассмотрены вопросы применения этого аппарата в различных практических областях. В соответствии с иерархией рассмотренных критериев можно построить дерево из интегралов Шоке 2-го порядка, подобное дереву, описанному, например, в работах [6,7]. Соответствующая структура представляет собой «вложенные» друг в друга интегралы Шоке.

Интегралы $C_{\mu_1}(g_1^1,...g_{21}^1), C_{\mu_2}(g_1^2,...g_{21}^2), C_{\mu_3}(g_1^3,...g_{21}^3),\dots C_{\mu_{13}}(g_1^{13},g_{14}^{13})$ служат для получения значений составных критериев $G_1,...,G_{13}$. Интеграл $C_{\mu 14}(G_1^{14},...G_{13}^{14})$ служит для агрегирования всех составных критериев. Результатом оценки является значение этого интеграла, полученное на основе значений входных критериев для веб-приложения. Выразим результат оценки Ω через соответствующие интегралы Шоке:

$$\Omega = C_{\mu_{14}}(C_{\mu_1}(g_1^1,...g_{21}^1), C_{\mu_2}(g_1^2,g_{21}^2), C_{\mu_3}(g_1^3,g_{13}^3),...C_{\mu 13}(g_1^{13},g_{14}^{13})) \quad (1)$$

Важно учитывать преимущества использования операторов агрегирования. При сравнении, например, с оценками, полученными с использованием средневзвешенного оператора, не учитываются резкие скачки в результатах, что может быть критично, если приложение или иной объект будет доходить до недопустимого предела. Операторы

агрегирования на основе интеграла Шоке устраняют этот недостаток т.к. учитывают зависимости между критериями.

Для работы с интегралом Шоке необходимо идентифицировать нечеткую меру. Для идентификации нечеткой меры можно воспользоваться методом максимизации энтропии при заданных ограничениях [3]. Метод заключается в том, что если мы обладаем какой-то частью знаний о поведении случайной величины, то к той части знаний, которая нам недоступна, следует относиться наиболее непредвзято, максимизируя энтропию этой величины с учётом имеющихся знаний. Для этого необходимо определить отношения предпочтения. Предпочтения формируются в зависимости от требований и желаемого результата.

Примером предпочтений в данной ситуации может служить ранжирование реализаций критериев из групп с G_1 по G_{13} по их важности с использованием документа OWASP Top 10 risks [2], в котором содержится список актуальных рисков безопасности веб-приложений в порядке убывания угрозы.

Исходя из этих предпочтений и, возможно дополнительных предпочтений, установленных экспертами, можно установить пороги безразличия т.е. установить ограничения на критерии при оценке [9]. Для идентификации нечетких мер $\mu_1,...,\mu_{14}$ применим специализированный пакет Kappalab [10], предназначенный для идентификации нечётких мер и для работы с интегралом Шоке в контексте теории полезности. Используя этот пакет, можно получить индексы взаимодействия и индексы Шепли [4], а также сами интегралы Шоке 2-го порядка.

Заключение: Современные веб-приложения являются не только полноправными конкурентами классических настольных приложений, но и расширяют границы использования, благодаря преимуществам облачных сервисов и кроссплатформенности. Однако, чем больше функционал, тем более сложен код, а чем сложнее код, тем больше он подвержен уязвимостям – что в итоге может привести к серьезным пробелам в безопасности. Поэтому оптимальным методом оценки будет использование операторов агрегирования, учитывающих зависимости и разрешающих некоторые неопределенности в данных, что является весомым преимуществом при оценке реальных веб-приложений. Не стоит забывать и про простую систему оценивания, в которой определены группы и сами критерии. В статье кратко описана модель оценки безопасности приложений на основе стандарта верификации безопасности приложения ASVS и применения нечеткого

интеграла Шоке. В продолжение данной статьи планируется провести работы по практическому применению и анализу предложенной модели.

Список литературы

1. Application Security Verification Standard Available at: https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project#tab=Downloads, accessed 27.02.2015.
2. OWASP Top 10 – 2013 The Ten Most Critical Application Security Risks Available at: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project#tab=OWASP_Top_10_for_2013, accessed 27.02.2015.
3. Сакулин С.А., Алфимцев А.Н. Формализация экспертных знаний об удобстве веб-страниц на основе агрегирования пользовательских критериев // Информационные технологии. № 6. 2014. С. 16-21.
4. Grabisch M., Orlovski S., Yager R. Fuzzy aggregation of numerical preferences// Handbook of Fuzzy Sets Series. 1998. Vol. 4: Fuzzy Sets in Decision Analysis, Operations Research and Statistics. P. 31-68.
5. Сакулин С.А., Алфимцев А.Н. К вопросу о практическом применении нечётких мер и интеграла Шоке // Вестник МГТУ им. Баумана. Сер. «Приборостроение». 2012. Спец. вып. № 4 «Компьютерные системы и технологии». С. 55-63.
6. Narukawa Y., Torra V. Twofold integral and Multi-step Choquet integral // Kybernetika-Praha. 2004. № 40(1). P. 39–50.
7. Тимонин М.В., Проблема оптимизации распределения ресурсов при планировании стратегии информационной безопасности // Безопасность информационных технологий. 2011. № 2. С. 90-96.
8. JaynesE.T. Informationtheoryandstatisticalmechanics // Phys. Rev. 1957. № 106. P. 620-630.
9. Сакулин С.А. К вопросу об идентификации параметров интеграла Шоке 2-го порядка // Вестник ИРГТУ. 2008. № 3(35). С. 205-208.
10. Grabisch M., Kojadinovic I., Meyer P. A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab. Available at: http://www.sciencedirect.com/science/article/pii/S0377221707002330, accessed 27.02.2015.