УДК 004.777.056.53 **doi: 10.32620/reks.2018.4.09**

А. Г. ТЕЦКИЙ

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина

ПРИМЕНЕНИЕ НЕЙРОСЕТЕЙ ДЛЯ ВЫБОРА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ НА ПРОНИКНОВЕНИЕ

Тестирование на проникновение проводится с целью обнаружения и дальнейшего устранения проблем безопасности Web-приложения. При проведении тестирования активно используются инструментальные средства, которые избавляют тестировщика от выполнения большого количества монотонных операций. Проблема выбора инструментальных средств состоит в том, что для тестирования одного и того же класса проблем безопасности существует некоторое количество похожих утилит, и неизвестно, какую утилиту лучше всего выбрать для конкретного случая. Такая проблема чаще всего встречается у начинающих тестировщиков, более опытные тестировщики используют собственные наборы утилит для поиска определенных проблем безопасности. Такие наборы формируются в процессе работы, и каждый тестировщик находит для себя наиболее подходящие утилиты. Целью статьи является создание метода, который поможет выбрать инструментальное средство для конкретного случая, основываясь на опыте экспертов в тестировании безопасности Web-приложений. Для достижения цели предлагается создать Web-сервис, который будет использовать нейросеть для решения задачи выбора. Данные для обучения нейросети в виде матрицы утилит и их критериев предоставляются экспертами в области тестирования безопасности Web-приложений. Для поиска наиболее подходящей утилиты должен быть сформирован вектор требований, т.е. пользователь сервиса должен указать критерии для поиска. В результате поиска пользователю показывается несколько утилит, наиболее подходящих по запросу. Также пользователь может сохранить результат своего выбора, если он отличается от предложенного. Таким способом множество обучающих примеров может быть расширено. Целесообразно иметь две нейросети, одна обучается только на данных от экспертов, вторая обучается на данных от экспертов и на данных пользователей, сохранивших свой выбор. Использование нейросетей позволяет реализовать соответствие нескольких наборов входных данных одному набору выходных данных. Описанный метод может использоваться для выбора программного обеспечения в различных сферах применения.

Ключевые слова: *тестирование на проникновение; Web-приложение; инструментальные средства; нейронные сети; кибербезопасность.*

Введение

Этап тестирования является одним из этапов жизненного цикла разработки программного обеспечения. Традиционно целью этого этапа является проверка корректности реализации функциональности разработчиками. В свою очередь, целью тестирования на проникновение является проверка возможностей использования дефектов разработки для компрометации системы. К сожалению, тестирование на проникновение не является обязательным этапом разработки программного обеспечения. Чаще всего тестирование на проникновение проводится для таких сфер применения, где взлом может нанести существенные убытки. Примерами таких сфер являются банковская сфера и бизнес. Взломав онлайн-банкинг, злоумышленники могут совершить нелегитимный перевод средств, а при взломе онлайн-магазина они могут получить платежную информацию, информацию о заказах и прочую информацию, которая может быть использована для получения прибыли.

Тестирование на проникновение является одним из способов поиска проблем безопасности Web-приложения [1]. Оно проводится на этапе эксплуатации программного обеспечения. Этот вид тестирования аналогичен взлому взлома приложения, однако тестирование на проникновение имеет другую цель. Результатом тестирования является создание отчета, с помощью которого разработчики могут исправить выявленные проблемы. В работе [2] показаны различия между различными типами тестирования на проникновение.

При проведении тестирования на проникновение широко используются различные инструменты автоматизации, что позволяет избежать ручного повторения монотонных операций. Наличие инструментов со схожей функциональностью вызывает

проблему выбора, поскольку не всегда понятно, какие инструменты лучше использовать в различных ситуациях для проведения тестов и выявления определенных классов уязвимостей. Тестировщик должен тратить время на поиск информации в дополнительных источниках, сравнивать инструменты и самостоятельно находить наиболее подходящие инструменты. Другая проблема заключается в том, что схожие инструменты могут давать разные результаты, не находить все уязвимости или вызывать ложные срабатывания [3].

В настоящее время информация об использовании инструментальных средств может быть почерпнута из различных источников, таких как онлайн-ресурсы [4], книги [5, 6], статьи [7]. Достаточно часто в статьях можно встретить сравнение результатов работы сканеров на специально созданных площадках с заранее известными уязвимостями. Например, в статье [8] проводилось сравнение многофункциональных сканеров, и было выявлено, что ни один из них не смог обнаружить уязвимости логики приложения. Поэтому всегда стоит помнить о том, что инструментальные средства не могут сами выполнить всю работу за тестировщика, они созданы для помощи тестировщику, но не могут его заменить.

Целью работы является создание метода выбора инструментальных средств тестирования на проникновение. Разработанный метод реализуется в виде Web-сервиса, который будет помогать выбирать утилиты в зависимости от вектора требований. Такой сервис является агрегатором с возможностью поиска.

1. Исходные данные

Обучение нейросети происходит на основе данных, полученных от экспертов в сфере тестирования Web-приложений на проникновение. В общем виде данные могут быть описаны следующим образом:

$$M = \begin{Bmatrix} m_{11}, m_{12}, \ldots, m_{1n}; \\ m_{21}, m_{22}, \ldots, m_{2n}; \\ \ldots; \\ m_{k1}, m_{k2}, \ldots, m_{kn} \end{Bmatrix}$$ – матрица значений критериев инструментальных средств,

m_{ij} – значение i-того критерия j-того средства,

k – размер множества критериев,

n – размер множества утилит.

Фрагмент данных, которые могут быть использованы для обучения, приведен в таблице 1. Эти данные были получены в результате проверки нескольких сканеров на тестовой площадке [9]. С помощью сканеров проверялись характерные для Web-приложений вектора атак.

Таблица 1

Соответствие критериев и сканеров

Критерий	Acunetix Web Vulnerability Scanner	IBM Rational AppScan	w3af
Server Side Java Script injection	1	0	0
Reflected Cross Site Scripting	1	1	1
Persistent Cross Site Scripting	1	1	1
DOM Cross Site Scripting	1	1	1
JSON Hijacking	0	1	0
Server-Side Includes Injection	0	1	1
Format String Attack	0	1	1
Code Injection	1	1	0
XML Injection	0	1	0
Forceful Browsing/Authentication Bypass	1	1	1
Privilege Escalation	0	1	0
Xml External Entity	1	1	0
Weak Session Identifier	0	1	1
Session Fixation	1	1	0
Cross Site Request Forgery	1	1	1

Как упоминалось ранее, множество обучающих примеров может быть расширено путем учета пользовательских мнений по поводу соответствия вектора требований и утилит, которые были предложены как наиболее подходящие по вектору требований. Такие данные нельзя использовать без предварительной проверки на наличие противоречивости. Под противоречивостью подразумевается соответствие одинаковых наборов входных данных различным наборам выходных данных. При обучении на противоречивых обучающих примерах результат работы нейросети будет непредсказуемым.

Для решения поставленной задачи используется нейросеть прямого распространения, для обучения с учителем используется метод обратного распространения ошибки [10].

2. Архитектура Web-сервиса

Для реализации задачи выбора инструментов для тестирования на проникновение предлагается создать Web-сервис, основной частью логики которого на стороне сервера будет нейронная сеть. Разработанный сайт будет помогать начинающим те-

стировщикам на проникновение в выборе инструментов. Для реализации нейронной сети предлагается использовать библиотеку FANN (Fast Artificial Neural Network) [11] в сочетании с традиционным программным стеком LAMP для Web-приложений (Linux + Apache + MySQL + PHP). Все используемые технологии бесплатны. Общий вид архитектуры веб-приложения показан на рисунке 1.

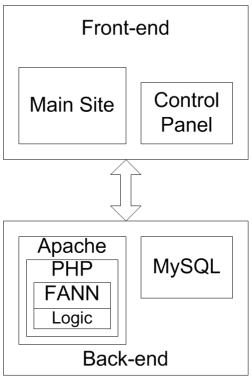

Рис. 1. Архитектура Web-сервиса

Основная часть сайта доступна всем пользователям, с ее помощью пользователь будет отвечать на вопросы, формируя тем самым вектор требований для поиска инструментов. Панель управления является закрытой, она доступна только администратору сайта. Используя панель управления, администратор выполняет следующие действия:

1. Добавление новых инструментов в систему. Технологии развиваются, появляются новые уязвимости и новые направления атак. Инструменты для обнаружения новых уязвимостей также должны быть обновлены для выявления текущих проблем безопасности Web-приложения. Когда выпускается новый инструмент, которого еще нет в системе, администратор сайта может добавить его в список инструментов.

2. Добавление новых характеристик инструментов. Каждый инструмент определяется определенным набором характеристик. Если добавленный инструмент обладает уникальными характеристиками, администратор может добавить их в общий список характеристик.

3. Определите значения характеристик инструмента. На основе таблицы исходных данных администратор вводит соответствующие значения каждой характеристики для каждого инструмента.

Информация об инструментах и их характеристиках хранится в базе данных. Эти данные используются при обучении нейронной сети. Кроме того, результаты выбора инструментов пользователями будут сохранены в базе данных, то есть пользователи могут сохранить результат выбора инструмента, если их выбор отличается от предложенного.

3. Пример реализации нейронной сети

Рассмотрим пример создания нейросети и ее обучения на данных, приведенных в таблице 1. Параметры нейронной сети:

– количество входных нейронов – 15 (соответствует количеству критериев утилит);

– количество выходных нейронов – 3 (соответствует количеству утилит);

– количество скрытых слоев – 2;

– количество нейронов в скрытых слоях – 9 и 6 соответственно.

Нет строгих правил по выбору количества нейронов скрытых слоев, указанные выше значения были получены с помощью правила геометрической пирамиды [12].

В таблице 2 показаны несколько примеров входных данных и результат работы нейросети. Результат состоит из трех чисел, что соответствует трём утилитам. Чем ближе значение к единице, тем больше уверенность нейросети в том, что утилита соответствует заданному вектору требований.

Таблица 2

Результаты работы нейросети

№ п/п	Входные данные	Результат
1	1; 1; 1; 1; 0; 0; 0; 1; 0; 1; 0; 1; 0; 1; 1	0,985; 0; 0,0007
2	1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0	0,9791; -0,1196; -0,0081
3	0; 1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0	0,9285; -0,137; 0,7764

В первом примере нейросеть выбрала только первую утилиту, так как входной вектор совпадал с одним из векторов обучающих данных. Во втором примере производился поиск по одному критерию, который есть только у первой утилиты, нейросеть также выбрала эту утилиту. В третьем примере производился поиск по критерию, присущему всем трем утилитам, однако, нейросеть не выбрала вторую утилиту. Такой результат был получен из-за недостаточного обучения нейронной сети. Для применения на практике нейронная сеть должна быть обучена на данных о 50-100 утилитах, показанный выше пример является лишь демонстрацией возможностей.

Заключение

В статье предложен метод решения задачи выбора инструментальных средств тестирования Web-приложений на проникновение. Описаны исходные данные для создания Web-сервиса, реализующего предложенный метод. Преимущество использования

нейронных сетей заключается в простоте реализации по сравнению с детерминированными алгоритмами, в качестве метрики используется количество строк кода. К недостаткам можно отнести необходимость экспериментального подбора параметров нейросети. Также сложностью является поиск данных для обучения ввиду высоких требований к экспертам, которые предоставляют обучающие данные для нейронной сети.

Направление дальнейших исследований заключаются в исследовании влияния параметров нейронной сети на результат ее работы.

Литература

1. Vieira, M. Using web security scanners to detect vulnerabilities in web services [Text] / M. Vieira, N. Antunes, H. Madeira // Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference. – IEEE, 2009. – P. 566-571.

2. Austin, A. One Technique is Not Enough: A Comparison of Vulnerability Discovery Techniques [Text] / A. Austin, L. Williams // 2011 International Symposium on Empirical Software Engineering and Measurement. – IEEE, 2011. – P. 97-106.

3. Awang, N. Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing [Text] / N. Awang, A. Manaf // Advances in Security of Information and Communication Networks. – Springer, Berlin, Heidelberg, 2013. – P. 230-239.

4. Kali Linux Penetration Testing Tools [Electronic resource] – Access mode: https://tools.kali.org. – 10.10.2018.

5. Metasploit: the penetration tester's guide [Text] / D. Kennedy, J. O'gorman, D. Kearns, M. Aharoni. – San Francisco, No Starch Press, 2011. – 328 p.

6. Engebretson, P. The basics of hacking and penetration testing: ethical hacking and penetration testing made easy [Text] / P. Engebretson. – Syngress, 2013. – 225 p.

7. Khari, M. An Overview of Black Box Web Vulnerability Scanners [Text] / M. Khari, N. Singh // International Journal of Advanced Research in Computer and Software Engineering. – IJARCSSE, 2014. – Vol. 4. – P. 1527-1535.

8. Doupé, A. Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners [Text] / A. Doupé, M. Cova, G. Vigna // Detection of Intrusions and Malware, and Vulnerability Assessment. – Springer, Berlin, Heidelberg, 2010. – P. 111-131.

9. Mirjalili, M. A survey on web penetration test [Text] / M. Mirjalili, A. Nowroozi, M. Alidoosti // Advances in Computer Science: an International Journal. – Los Alamitos, CA, 2014. – Vol. 3, No. 6. – P. 107-121.

10. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечёткие системы [Текст] / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. : Горячая линия – Телеком, 2006. – 452 с.

11. Fast Artificial Neural Network Library (FANN) [Electronic resource]. – Access mode: http://leenissen.dk/fann/html/files/fann-h.html. – 10.10.2018.

12. Masters, T. Practical neural network recipes in C++ [Text] / T. Masters. – Morgan Kaufmann, 1993. – 493 p.

References

1. Vieira, M., Antunes, N., Madeira, H. Using web security scanners to detect vulnerabilities in web services. *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference*, IEEE, 2009, pp. 566-571.

2. Austin, A., Williams, L. One Technique is Not Enough: A Comparison of Vulnerability Discovery Techniques. *2011 International Symposium on Empirical Software Engineering and Measurement*, IEEE, 2011, pp. 97-106.

3. Awang, N., Manaf, A. Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing. *Advances in Security of Information and Communication Networks*, Springer, Berlin, Heidelberg, 2013, pp. 230-239.

4. *Kali Linux Penetration Testing Tools*. Available at: https://tools.kali.org (accessed 10.10.2018).

5. Kennedy, D., O'gorman, J., Kearns, D., Aharoni, M. *Metasploit: the penetration tester's guide*. San Francisco, No Starch Press, 2011. 328 p.

6. Engebretson, P. *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy*. Syngress, 2013. 225 p.

7. Khari, M., Singh, N. An Overview of Black Box Web Vulnerability Scanners. *International Journal of Advanced Research in Computer and Software Engineering*, IJARCSSE, 2014, vol. 4, pp. 1527-1535.

8. Doupé, A., Cova, M., Vigna, G. Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners. *Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, Berlin, Heidelberg, 2010, pp. 111-131.

9. Mirjalili, M., Nowroozi, A., Alidoosti, M. A survey on web penetration test. *Advances in Computer Science: an International Journal*, Los Alamitos, CA, 2014, vol. 3, no. 6, pp. 107-121.

10. Rutkovskaya, D., Pilin'skii, M., Rutkovskii, L. *Neironnye seti, geneticheskie algoritmy i nechetkie sistemy* [Neural networks, genetic algorithms and fuzzy systems]. Moscow, "Goryachaya liniya – Telekom" Publ., 2006. 452 p.

11. *Fast Artificial Neural Network Library (FANN)*. Available at: http://leenissen.dk/fann/html/files/fann-h.html (accessed 10.10.2018).

12. Masters, T. *Practical neural network recipes in C++*. Morgan Kaufmann Publ., 1993. 493 p.

Поступила в редакцию 10.10.2018, рассмотрена на редколлегии 12.12.2018

ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖ ДЛЯ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТЕСТУВАННЯ WEB-ДОДАТКІВ НА ПРОНИКНЕННЯ

А. Г. Тецький

Тестування на проникнення проводиться з метою виявлення і подальшого усунення проблем безпеки Web-додатка. При проведенні тестування активно використовуються інструментальні засоби, які позбавляють тестувальника від виконання великої кількості монотонних операцій. Проблема вибору інструментальних засобів полягає в тому, що для тестування одного і того ж класу проблем безпеки існує деяка кількість схожих утиліт, і невідомо, яку утиліту найкраще вибрати для конкретного випадку. Така проблема найчастіше зустрічається у тестувальників – початківців, більш досвідчені тестувальники використовують власні набори утиліт для пошуку певних проблем безпеки. Такі набори формуються в процесі роботи, і кожен тестувальник знаходить для себе найбільш підходящі утиліти. Метою статті є створення методу, який допоможе вибрати інструментальний засіб для конкретного випадку, ґрунтуючись на досвіді експертів в тестуванні безпеки Web-додатків. Для досягнення мети пропонується створити Web-сервіс, який буде використовувати нейромережу для вирішення завдання вибору. Дані для навчання нейромережі у вигляді матриці утиліт і їх критеріїв надаються експертами в області тестування безпеки Web-додатків. Для пошуку найбільш підходящої утиліти повинен бути сформований вектор вимог, тобто користувач сервісу повинен вказати критерії для пошуку. В результаті пошуку користувачеві показується кілька утиліт, найбільш підходящих за запитом. Також користувач може зберегти результат свого вибору, якщо він відрізняється від запропонованого. Таким способом множину навчальних прикладів може бути розширено. Доцільно мати дві нейромережі, одна навчається тільки на даних від експертів, друга навчається на даних від експертів і на даних користувачів, що зберегли свій вибір. Використання нейромереж дозволяє реалізувати відповідність декількох наборів вхідних даних одному набору вихідних даних. Описаний метод може використовуватися для вибору програмного забезпечення в різних сферах застосування.

Ключові слова: тестування на проникнення; Web-додаток; інструментальні засоби; нейронні мережі; кібербезпека.

APPLYING OF NEURAL NETWORKS FOR SELECTING THE TOOLS FOR PENETRATION TESTING OF WEB APPLICATIONS

A. G. Tetskyi

Penetration testing is conducted to detect and further to fix the security problems of the Web application. During testing, tools are actively used that allows to avoid performing a large number of monotonous operations by the tester. The problem with selecting the tools is that there are a number of similar tools for testing the same class of security problems, and it is not known which tool is most suitable for a particular case. Such a problem is most often found among novice testers, more experienced testers use their own sets of tools to find specific security problems. Such kits are formed during the work, and each tester finds the most suitable tools for him. The goal of the paper is to create a method that will help to choose a tool for a particular case, based on the experience of experts in security testing of Web applications. To achieve the goal, it is proposed to create a Web service that will use the neural network to solve the problem of choice. Data for training a neural network in the form of a matrix of tools and their criteria are provided by experts in the field of security testing of Web applications. To find the most suitable tool, a vector of requirements should be formed, i.e. the user of service must specify the criteria for the search. As a result of the search, several most suitable for the request tools are shown to the user. Also, the user can save the result of his choice, if it differs from the proposed one. In this way, a set of learning examples can be extended. It is advisable to have two neural networks, the first one is trained only on data from experts; the second one is trained on data from experts and on data of users who have retained their choice. The usage of neural networks allows to realize correspondence between several input data sets to the one output data set. The described method can be used to select software in various applications.

Keywords: penetration testing; web application; tools; neural networks; cybersecurity.

Тецкий Артём Григорьевич – ассистент кафедры компьютерных систем, сетей и кибербезопасности Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина.

Tetskyi Artem Grygorovych – Assistant Lecturer of Dept. of Computer Systems, Networks and Cybersecurity, National Aerospace University "Kharkov Aviation Institute", Kharkov, Ukraine, e-mail: a.tetskiy@csn.khai.edu.