Analysis of the Possibilities of Unauthorized Access in Content Management Systems Using Attack Trees

Artem Tetskyi, Vyacheslav Kharchenko, Dmytro Uzun

National Aerospace University "KhAI", Kharkiv, Ukraine
{a.tetskiy, v.kharchenko, d.uzun}@csn.khai.edu

Abstract. The reasons for attacks on content management systems are considered. Frequent attack scenarios for obtaining unauthorized access are investigated. The method for assessing the probability of a successful attack on a content management system is proposed. Described method uses the attack tree, audit results, source code analysis results, and statistical data. Combinations of basic events for high probability of successful attack are defined. The differences of the proposed method from existing methods of security assessment are shown.

Keywords: attack, content management system, attack tree, unauthorized access, cyber security, web applications.

1 Introduction

1.1 Motivation

Demand for modern organizations to provide information in order to increase sales or provide services implies the creation of information resources on the Internet. Content management systems are often used to create sites [1].

Content management system (CMS) is software that allows to edit web pages and to create websites based on them. Examples of such systems are Wordpress, Joomla and others. Similar systems have found application in education, for example, MOODLE – a management system for learning content. Such systems have become widely known due to ease of use, the number of installations can be measured in millions of copies. A particular feature of such systems is the modular architecture, which makes it possible to manage the functionality of the site by installing the necessary modules. A critical vulnerability in the module can endanger all sites that use this module, so attackers can hack many of these sites in the same scenario [2]. The growing popularity of content management systems makes them an interesting target for intruders. Among the features of using content management systems in the aspect of information security are the following:

– High prevalence and a large community of users, which can detect vulnerabilities before attackers and pass information to system developers for the release of the patch.

- Regular updates of the kernel and system modules, in which previously discovered vulnerabilities can be eliminated but it may contain new vulnerabilities.
- Not all site administrators install updates. One of the reasons for this is the possible incompatibility of the kernel and the new version of the module or the incompatibility of the modules with each other.
- A wide range of applications is due to the diverse functionality of the modules that can be installed. Any developer can create his own module and make it available for installation to the entire community. It is not known what vulnerabilities this module can contain.
- The use of content management systems in electronic business also attracts intruders. Hacking an online store, an attacker gets access to information that he can sell to a competing online store. Having cracked the online exchanger of electronic money, an attacker can access the accounts of various payment systems and transfer money to arbitrary accounts.

Thus, content management systems provide a wide field for information security activities [3].

1.2 Related Works Analysis

To investigate attack scenarios of web applications, it is suggested to use attack tree analysis. The method of analysis of trees (failures, attacks) is applied in such spheres as aviation, nuclear industry, military industry, etc. In the field of information technology, attack trees can be used to visualize possible ways of attacking various components of a computer system. In [4] the attack tree was used to analyze attacks in the corporate network. The proposed method was based on system log files analysis. In source [5] attack-defense tree based security assessment method for vehicular ad hoc network was described. The analysis of attack trees has found application in the rapidly developing Internet-of-Things area. In the paper [6] an approach to characterizing malicious and unintentional insider threats on the Internet-of-Things was presented. An example of the attack tree was provided in [7] to access the victim's email. In [8] security issues specific to web applications, possible causes of attacks and their consequences were discussed. As the information was provided in text form, rather than in graphic form, the visibility of such information was lower. The proposed approach is to apply a known method of analysis to real-world scenarios of web application attacks.

1.3 Goal

Understanding attack scenarios allows to apply the necessary methods for protecting a web application. The results of the web audit give understanding of how much the administrator takes care of the security of his account. Analyzing the source code, unlike testing for black box penetration, allows to detect much more possible vulnerabilities. Statistics provides information about how often specific attack scenarios have been successful. Proceeding from the foregoing, the problem of determining the prob-

ability of a successful attack P based on investigation of possible attack scenarios, audit results, source code analysis and statistical data arises.

$$P = f(scenarios, audit, src, stats) \quad (1)$$

Based on research, a method for assessing the probability of the successful attack of a particular site is developing. Conducting either an audit or a penetration testing can't answer this question, because penetration testing does not cover the problem of non-compliance with security policies, and the audit does not cover the vulnerabilities of the source code. Combining the results of these processes can give a more objective assessment, because the attacks are covered, which can be successful not only because of the exploitation of the vulnerabilities but also because of the violation of security policies.

2 Investigation of Possible Attack Scenarios

First, you need to determine the main event and investigate possible attack scenarios.

The main event is a successful web application attack. A successful attack involves gaining unauthorized access to functions that are only available to the administrator from the control panel. The attack options are divided into groups and are presented below:

- attacks with password stealing;
- attacks with the brute force of password;
- attacks on vulnerabilities in the content management system.

A fragment of the tree that displays the main event and three ways to achieve it is shown in the Fig. 1.

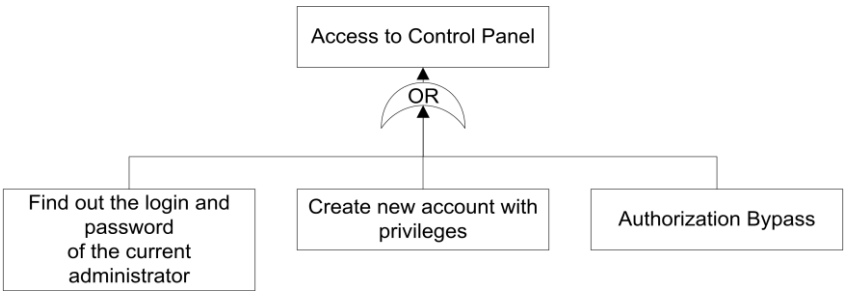

Fig. 1. Fragment of attack tree with main event

The event "Find out the login and password of the current administrator" means attacking with stealing or selecting a password, i.e. the attacker recognized the administrator credentials. Events "Create new account with privileges" and "Authorization Bypass" are based on the exploitation of vulnerabilities in the content management system.

Attacks with password stealing. The presence of critical vulnerabilities in the content management system increases the likelihood of a successful attack, but in some cases, attackers do not use these vulnerabilities. This is due to the fact that there are several ways to find out the administrator's login and password. For example, an attacker can spy on them while authorizing, being close to the administrator. They can be read on a sticker that lies under the keyboard or is attached to the monitor. They can be intercepted with an unencrypted connection or kidnapped from a mailbox. Most of the above options can be implemented due to a low level of knowledge of the user in the field of information security. To reduce the likelihood of successful use of these scenarios, the requirements of the security policy that prohibit the storage of the password in clear form on digital and non-digital media must be observed.

Attacks with the brute force of a password. An important element of the security policy is the complexity of the password, the prohibition of the use of dictionary passwords, the need for periodic password changes, the prohibition of re-use of passwords, the prohibition of the use of the same passwords on different sites. Attackers can try to crack a password, the harder the password, the less likely the successful cracking password. Using the same password on different sites can have negative consequences – compromising the database of one site reveals the credentials of users who may be the same on another site.

A tree fragment that depicts how to compromise credentials is shown in Figure 2.

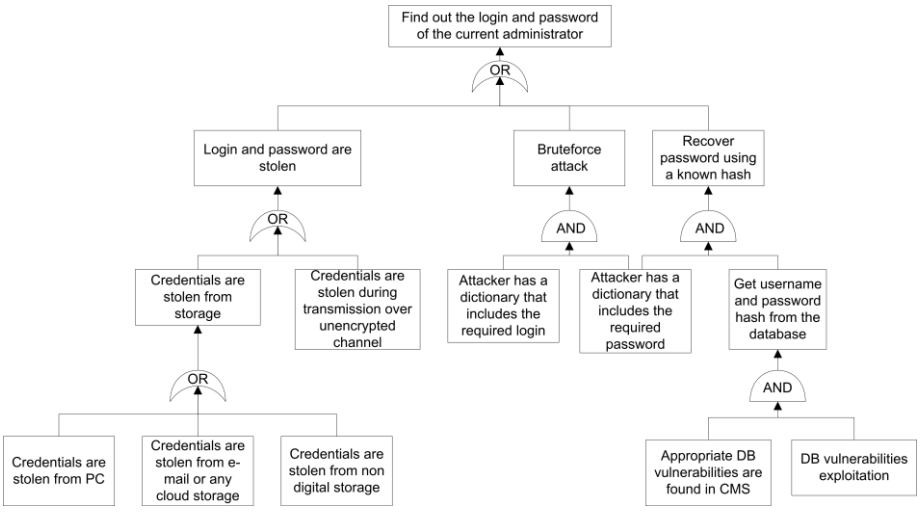

Fig. 2. Ways of compromising credentials

Attacks on vulnerabilities in the content management system. Defects of the program code allow attackers to penetrate into those parts of the content management system, which should be accessible only to the administrator. There are several classes of vulnerabilities that allow performing administrative functions, among them the following can be distinguished:

– incorrect processing of access rights to objects;
– enabling the remote file;
– running arbitrary code.

Some vulnerabilities can be part of a complex hacking scenario. For example, invalid sanitization of input data allowed to change the logic of the query to the database, there was a compromise of the login and hash of the password. The next step is to crack a password for a known hash – this is another way to get the administrator password.

The content management system may contain vulnerabilities that allow to bypass the authorization mechanism and to gain access to administrator functions. It is difficult to distinguish common cases, so experts will need to assess the possibility of existence of such vulnerabilities in the system under investigation. The tree fragment for the "Authorization Bypass" event is shown in Figure 3.

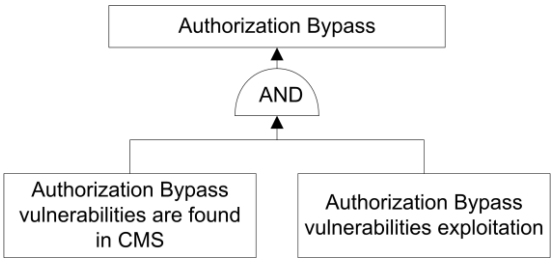

Fig. 3. Fragment tree for the "Authorization Bypass" event

Some vulnerabilities can allow to raise the level of a regular user to an administrator or add a new administrator to the database directly. There are cases when an attacker becomes aware of data to connect to the database (user name and password of the database user). If it is possible to connect directly to the database or access to a web-based database management interface is present, an attacker will be able to access the database and make changes. The creation of a new administrator by an attacker is shown in Figure 4.

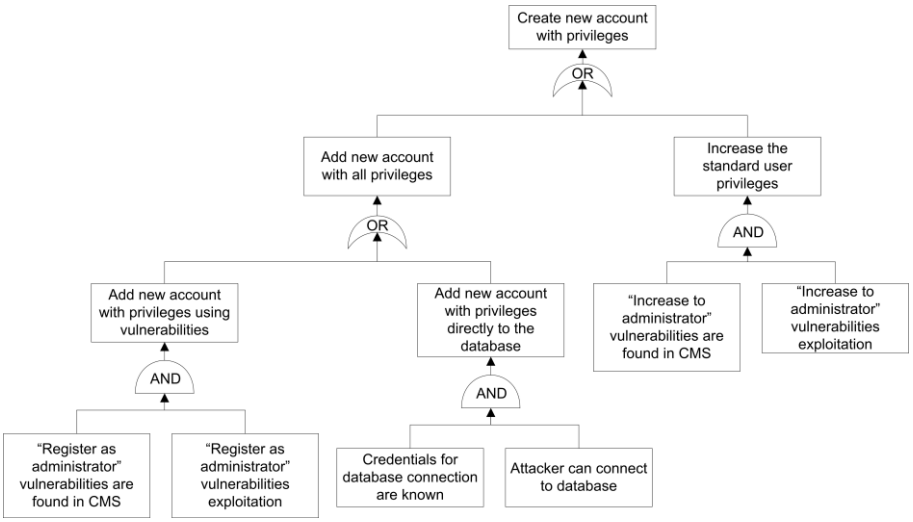

Fig. 4. Ways to create a new administrator by an attacker

Trees were chosen because of their representativeness and the ability to use quantitative and qualitative values to determine the probabilities of basic events.

It cannot be said that all possible scenarios were shown above, since some of them can be partially combined, some are unique to specific content management systems. The constructed tree does not take into account the possibility of using two-factor authentication for access to the control panel. It is also assumed that there is no protection against brute force of logins and passwords.

3 Determining the Probability of Obtaining Unauthorized Access

The next step is to parameterize the attack tree. The sources of the parameters are web audit data, statistical data, and source code analysis results. To determine the probability of each initial event, rules should be established, through which the results of audit and penetration testing [9] are converted into parameters of the created tree. Because there is no representative sample of the occurrence of events of attacks on certain components of the system, and it is practically impossible to determine the numerical values of probability, therefore it is proposed to use three levels for assessment – low, medium, high. The probability of events combined by the AND operation is determined by the minimum probability value, for the OR operation the maximum value is selected, that is:

$$P(A\ AND\ B) = min(P(A), P(B)) \tag{2}$$

$$P(A\ OR\ B) = max(P(A), P(B)) \tag{3}$$

Let’s consider the definition of probabilities of events associated with the stealing of credentials. The first event is "Credentials are stolen from PC". A low level is assigned by the expert, provided that the credentials are not stored on the computer (in documents, in the browser, etc.). The middle level is assigned if the login and password are stored in the file without the appropriate notations explaining the purpose of these credentials. A high level is assigned when storing credentials in the browser, or in documents explaining to which system these credentials belong.

In the process of audit, it is need to check the complexity of the used passwords and verify that commonly used logins are not being used. To assess the complexity of the password, existing services can be used, for example, www.passwordmeter.com. An example of password protection policies is shown in [10].

To determine the probability of events associated with the presence and possibility of exploiting vulnerabilities in the system under consideration, an integrated approach can be applied. It consists in that, first of all, the system should be checked for known vulnerabilities. For some vulnerabilities, exploits can be found in open access. At the next stage, penetration testing [11] can be performed using various tools [12]. But the best way [13] to find vulnerabilities in the code is to analyze the source code.

Similar rules can be developed for each initial event.

Define the parameter sets for the high probability of the occurrence of the main event. As a result of the construction of the tree, there were 16 basic events at the bottom level, from the probability of which the probability of occurrence of the main event depends. In Table 1 high probability is denoted by the symbol "1", the symbol "*" denotes values that do not affect the result of calculations. In some cases, the probability of the main event is determined by the probability of only one basic event. This is due to the fact that OR elements dominate in the constructed tree, in particular, in the upper part of the tree.

Table 1. The values of the probabilities of basic events with a high probability of the main event

Credentials are stolen from PC	Credentials are stolen from e-mail or any cloud storage	Credentials are stolen from non digital storage	Credentials are stolen during transmission over unencrypted channel	Attacker has dictionary that includes the required login	Attacker has a dictionary that includes the required password	Appropriate DB vulnerabilities are found in CMS	DB vulnerabilities exploitation	Authorization Bypass vulnerabilities are found in CMS	Authorization Bypass vulnerabilities exploitation	“Register as administrator” vulnerabilities are found in CMS	“Register as administrator” vulnerabilities exploitation	Credentials for database connection are known	Attacker can connect to database	“Increase to administrator” vulnerabilities are found in CMS	“Increase to administrator” vulnerabilities exploitation	Result
1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1
*	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	1
*	*	1	*	*	*	*	*	*	*	*	*	*	*	*	*	1
*	*	*	1	*	*	*	*	*	*	*	*	*	*	*	*	1

*	*	*	*	1	1	*	*	*	*	*	*	*	*	*	*	1
*	*	*	*	*	1	1	1	*	*	*	*	*	*	*	*	1
*	*	*	*	*	*	*	*	1	1	*	*	*	*	*	*	1
*	*	*	*	*	*	*	*	*	*	1	1	*	*	*	*	1
*	*	*	*	*	*	*	*	*	*	*	*	1	1	*	*	1
*	*	*	*	*	*	*	*	*	*	*	*		*	1	1	1

Using the Table 1, combinations of events that are highly likely to lead to a successful attack of the web application can be identified.

For high, medium and low levels, membership functions based on statistics can be described. Let's assume that with a low probability of 10 attacks, success was from 0 to 3 attacks inclusive, with a medium probability from 4 to 7 attacks were successful, with a high probability 8 or more attacks were successful. Thus, it is possible to define a linguistic variable [14] with the name x = " probability of successful attack". The universal set is U = [0,10]; the term-set is T = {"low", "medium", "high"} with such membership functions (u ε U):

$$\mu_{low}(u) = \frac{1}{1 + \left(\frac{u - 1.5}{2}\right)^{16}} \tag{4}$$

$$\mu_{medium}(u) = \frac{1}{1 + \left(\frac{u - 5.5}{2}\right)^{16}} \tag{5}$$

$$\mu_{high}(u) = \frac{1}{1 + \left(\frac{u - 9.5}{2}\right)^{16}} \tag{6}$$

Formulas 4-6 use a generalized bell-shaped function [15], which is based on three parameters. A graphical representation of the membership functions described above is shown in Figure 5.

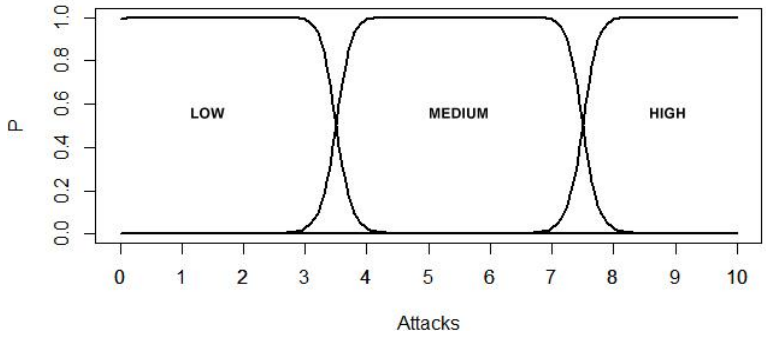

Fig. 5. Graphical representation of membership functions

This approach allows to assess how easily unauthorized access to the administrator functions of a particular web application can be obtained. A set of activities aimed at researching the source code and compliance with security policies, allows to determine the probabilities of the basic events of the constructed tree.

The difference from the existing methods of assessing the security of a web application, examples of which are given in [16, 17], is the following:

- Binding to real attack scenarios. This gives maximum proximity to the actions of an attacker when attacking a web application.
- Use of audit results. This allows you to consider attack scenarios that are not associated with the vulnerabilities of the target system.
- Lack of binding to detected vulnerabilities. When assessing the probability of a successful attack, known vulnerabilities (from bases such as NVD) are not taken into account, but experts can take into account the number and criticality of vulnerabilities in assessing the probabilities of basic events.

The disadvantage of this approach is the cost of conducting web audit and researching the source code.

4 Conclusions and Future Work

The aim of the research, which consists in investigating attack scenarios on the web application and determining the probability of a successful attack, was achieved by constructing and analyzing the attack tree. The research was based on the assumption that only the most common attack scenarios [18] are collected in the tree, and it is impossible to predict all attack scenarios that can be exploited by intruders.

The direction of further research is related to the identification of the most dangerous attacks and the creation of a method for selecting countermeasures.

References

1. López, J. M., Pascual, A., Masip, L., Granollers, T., Cardet, X.: Influence of web content management systems in web content accessibility. In IFIP Conference on Human-Computer Interaction (pp. 548-551). Springer, Berlin, Heidelberg (2011). doi:10.1007/978-3-642-23768-3_79
2. Slider Revolution Plugin Critical Vulnerability Being Exploited, https://blog.sucuri.net/2014/09/slider-revolution-plugin-critical-vulnerability-being-exploited.html, last accessed 2018/01/26.
3. Rehman H., Nazir M., Mustafa K.: Security of Web Application: State of the Art. In: Kaushik S., Gupta D., Kharb L., Chahal D. (eds) Information, Communication and Computing Technology. ICICCT 2017. Communications in Computer and Information Science, vol. 750, pp. 168-180. Springer, Singapore (2017). doi:10.1007/978-981-10-6544-6_17
4. Poolsapassit N., Ray I.: Investigating Computer Attacks Using Attack Trees. In: Craiger P., Shenoi S. (eds) Advances in Digital Forensics III. DigitalForensics 2007. IFIP — The

International Federation for Information Processing, vol. 242, pp. 331-343. Springer, New York, NY (2007). doi:10.1007/978-0-387-73742-3_23
5. Du, S., Zhu, H.: Security assessment via attack tree model. In Security Assessment in Vehicular Networks (pp. 9-16). Springer, New York, NY (2013). doi:10.1007/978-1-4614-9357-0_2
6. Kammüller F., Nurse J.R.C., Probst C.W.: Attack Tree Analysis for Insider Threats on the IoT Using Isabelle. In: Tryfonas T. (eds) Human Aspects of Information Security, Privacy, and Trust. HAS 2016. Lecture Notes in Computer Science, vol. 9750, pp. 234-246. Springer, Cham (2016). doi:10.1007/978-3-319-39381-0_21
7. Nagaraju, V., Fiondella, L., Wandji, T.: A survey of fault and attack tree modeling and analysis for cyber risk management. In Technologies for Homeland Security (HST), 2017 IEEE International Symposium on (pp. 1-6). IEEE (2017). doi:10.1109/THS.2017.7943455
8. Lepofsky, R.: The manager's guide to web application security: a concise guide to the weaker side of the web. Apress (2014). doi:10.1007/978-1-4842-0148-0
9. Messier R.: What Is Penetration Testing? In: Penetration Testing Basics (pp. 1-11). Apress, Berkeley, CA (2016). doi:10.1007/978-1-4842-1857-0_1
10. Password Protection Policy, https://www.sans.org/security-resources/policies/general/pdf/password-protection-policy, last accessed 2018/01/26.
11. Shah, S., Mehtre, B. M.: An overview of vulnerability assessment and penetration testing techniques. Journal of Computer Virology and Hacking Techniques, vol. 11(1), pp. 27-49 (2015). doi:10.1007/s11416-014-0231-x
12. Awang N.F., Manaf A.A.: Detecting Vulnerabilities in Web Applications Using Automated Black Box and Manual Penetration Testing. In: Awad A.I., Hassanien A.E., Baba K. (eds) Advances in Security of Information and Communication Networks. Communications in Computer and Information Science, vol. 381, pp. 230-239. Springer, Berlin, Heidelberg (2013). doi:10.1007/978-3-642-40597-6_20
13. Doupé A., Cova M., Vigna G.: Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners. In: Kreibich C., Jahnke M. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2010. Lecture Notes in Computer Science, vol. 6201, pp. 111-131. Springer, Berlin, Heidelberg (2010). doi:10.1007/978-3-642-14215-4_7
14. Zadeh, L. A.: The concept of a linguistic variable and its application to approximate reasoning—I. Information sciences, vol. 8(3), pp. 199-249 (1975). doi:10.1016/0020-0255(75)90036-5
15. Zhao J., Bose B.K.: Evaluation of membership functions for fuzzy logic controlled induction motor drive. In: Proceeding Annual Conference of the IEEE Industrial Electronics Society, vol. 1, pp. 229-234 (2002). doi: 10.1109/IECON.2002.1187512
16. Yu X., Jiang G.: A Web Security Testing Method Based on Web Application Structure. In: Huang Z., Sun X., Luo J., Wang J. (eds) Cloud Computing and Security. Lecture Notes in Computer Science, vol. 9483, pp. 244-258. Springer, Cham (2015). doi:10.1007/978-3-319-27051-7_21
17. Zech, P., Felderer, M., Breu, R.: Knowledge-based security testing of web applications by logic programming. International Journal on Software Tools for Technology Transfer, pp.1-26. Springer, Berlin, Heidelberg (2017). doi:10.1007/s10009-017-0472-3
18. Most Common Attacks Affecting Today's Websites, https://blog.sucuri.net/2014/11/most-common-attacks-affecting-todays-websites.html, last accessed 2018/01/26.