Syntax Literate: Jurnal Ilmiah Indonesia p–ISSN: 2541-0849
e-ISSN: 2548-1398
Vol. 7, No. 8, Agustus 2022

CONCURRENT IMPLEMENTATION OF WAF AND HARDENING BROKEN AUTHENTICATION TO SECURE WEB APPLICATION

Dimas Charis Suryo Nugroho, Muhammad Fadlan Hidayat, Benfano Soewito
Universitas Bina Nusantara, Jakarta, Indonesia
Email: dimas.nugroho008@binus.ac.id, muhammad.hidayat003@binus.ac.id, bsoewito@binus.ed

Abstract

Web Application Security is considered crucial in the era of rapid technology development, following with recent rapid development of artificial intelligence architecture, virtual reality and the internet of things, one of important node which is a web application that is connected to others node is needed to be protected from cyber-attack attempts scenarios. These cyber-attack scenarios are currently growing and evolve nowadays and cause a lot of losses in various countries, such as economic loss, privacy loss, safety loss and etc. Some research studies have developed web application fire-wall using artificial intelligence and show that using web application firewall are effective to tackle some of cyber-attack scenarios attempt However, the authors want to revisit and con duct several experiments on this web application firewall and also In this research the authors study and communicate proposed cyber security method through an experiment of concurrent implementation from Web Application Firewall (WAF) and Hardening Broken Authentication (HBA) method to secure OWASP 2017 most happening and popular two cyber-attack types namely Injection and Broken Authentication. As cyber-attacks are increased and evolving against recent web application firewalls, the authors successfully secure 14 out of 16 cyber-attack scenarios, and 11 out of 16 cyber-attack scenarios are perfectly secured by the concurrent implementation of web application (WAF) and Hardening Broken Authentication (HBA) without significant increased average network access from 51 milliseconds to 61 milliseconds according to authors experiment with fair same internet connection and devices compared to other security methods in this experiment.

Keywords: Web Application Firewall, Hardening Broken Authentication, Cyber Security, OWASP 2017, Injection, Broken Authentication.

Introduction

Data and information are essential for an organization to operate in modern society and in most cases, the organization's data and information is managed on the web application. There are a massive amount of data and information shared and stored on the web application through the internet. Forbes.com reported there are 2.5 quintillion bytes of data produced each day on a worldwide scale in addition statista.com also summarize in January 2021 there were 4.66 billion active internet users

How to cite: Dimas Charis Suryo Nugroho, et al (2022). Concurrent Implementation of WAF and Hardening Broken Authentication To Secure Web Application. *Syntax Literate; Jurnal Ilmiah Indonesia. 7* (8).
E-ISSN: 2548-1398
Published by: Ridwan Institute

worldwide [1] and for Indonesia, there were 202.6 million internet users. The growth of Indonesia internet users was 16% or 27 million users between 2020 until 2021 [2].

The huge growth of information reported is also on par with the number of penetrations and cyber-crime attempts, these penetrations attempt has a various purpose, several penetrations are meant to take benefit, misuse the data for wrongdoing, and attempts to prove themselves, or have fun with the information. According to the data compiled by Indonesia State Cyber and Code Agency (BSSN) in a recap of web defacement, there were 88.414.296 attempts from 1 January 2020 to 12 April 2020, these number of cyber-attacks keep increasing due to covid-19 policy, especially work and learn from home mandatory policy [3]. The huge growth of cyber-attack reported in web applications resulting importance in the web security and web security is considered critical and dealing with web security and cyber-attack attempts are progressively challenging [4].

In this research, the authors conduct four experiments on web security based on several top scenarios arranged by the Open Web Application Security Project 2017 [5]. The OWASP 2017 reported the first rank of cyber-attack was launched was injection and the second rank is broken authentication. The research conducts an experiment to secure the first rank and second rank of OWASP 2017 cyber-attack scenario with a total of 16 scenarios with 8 scenarios of each type using the concurrent implementation of web application firewall (WAF) and Hardening Broken Authentication (HBA) method. The authors use security assessment methodology in the guidance of Resta and Affandy study in 2019. In this cyber-security research also complemented with network access performance evaluation in each security attack scenarios. The authors also conduct an upgrade in hardening broken authentication methods in Ibor study [6] and provide a security novelty package to secure web applications.

This research article describes and communicates an experiment of securing web applications from cyber-attack of OWASP 2017 1st rank (Injection) and 2nd rank (Broken Authentication) using concurrent implementation Web Application Firewall and Hardening Broken Authentication method. The paper is categorized as follows. Section 1 is explained the introduction of information and cyber security current development, section 2 describes related re-search of cyber security methods to handle cyber-attack, section 3 describes the experiment methodology of the web security model and attack scenario conducted, sec-tion 4 informs the experiment result and section 5 reveals the insight from experiment results.

Related Researchs

In this section the authors discuss about how crucial protecting one of important ICT infrastructure which is web application, discuss popular cyber-attack scenario of Open Web Application Security (OWASP) and summarize several recent research on Web Application Firewall (WAF).

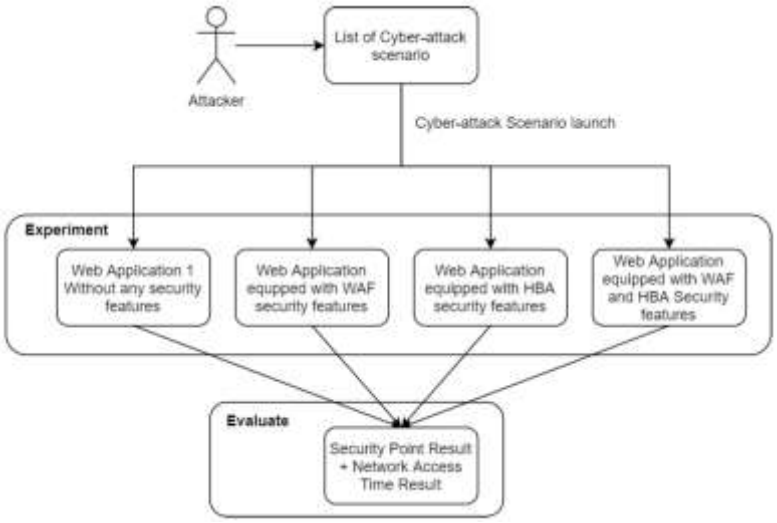

Figure 1
Experiment Methodelogy

The web applications in 2022 without a doubt is a crucial instrument for an organization. Web applications are commonly used to provide services, set an organizational mechanism or rules, media data interaction and etc. Even though this instrument development is in rapid pace there are also defect of the instrument that keep evolve and grow over the year along with the instrument development, one of the most common defect of web applications and also crucial is the security against cyber-attack [7].

The OWASP is a worldwide community focused on improving the security of application, all of OWASP tools and documents are open to anyone who is interested in developing secure applications, The top 10 popular cyber-attack summarized by OWASP 2017 list as well as thousands of other known vulnerabilities can be detected with web application security testing tools automatically or by security experts manually [5], [8]. The first ranked cyber-attack in OWASP 2017 study is injection type where this cyber-attack injection type is allowing malicious attackers to manipulate web server database, which can cause dangerous scenarios such as stealing the data, manipulating information, or damaging the information [9]. The second cyber-attack concerned by this research is OWASP 2017 broken authentication cyber-attack type. Where this cyber-attack type is related to exploit weakness of web system mechanism related to authentication and session so the attacker can ignore passwords, token and session, these cyber-attacks are able to maintain their illegitimate user access temporarily or permanently basis in worst case take over the account [10]–[12].

Answering a huge amount and evolving cyber-attack in the above paragraph, several recent research attempts to improve web application firewall from signature base technique to machine learning firewall approaches. Web application firewall used to guard at application-level exploitation or cyber-attack scenarios, web application firewall is popular Intrusion Detection System (IDS) to protect IT Infrastructure [13],

the idea of this web application firewall is functionally to detect and drop potential dangerous cyber-attack through analyze HTTP requests and drop the potential cyber-attack before it's happening, web application firewall is not only used to protect web application but it also protecting several node that connected to web application such as web server, databases, application programming interfaces (APIs) node such as recent popular smart door an internet of things crucial instrument from cyber-attack [14].

A signature-based web application firewall commonly make uses of security model base on cyber-attack database are defined or how policies are implemented. Two main security models are used: the negative security model and the positive security model. Web application firewall adopting negative security model allows all traffic to pass unless it matches defined cyber-attack, in which case traffic is blocked. If traffic does not match the rules, the security model allows traffic to pass. The negative security model has typically implemented the use of a signature-based approach [15].

A signature-based web application firewall (WAF) is basically a search-based technique approach from existing blocked cyber-attack or policies. Research by applet et al shows that this web application firewall signature base or negative-positive model security may be insufficient since hacker may evolve their attacks to work around existing security policies or rules, this statement is proven by applet et al with successfully bypass the current state-of-the-art web application firewall signature base [16], [17].

Currently, hackers evolve their attack using generative adversarial algorithms to bypass the web application firewall. Several research on machine learning base firewall also implemented and designed. Machine learning web application firewall propose by Ceccato et al [18], a clustering method for detecting SQL Injection attacks against web application services. This algorithm learns from the queries that are processed inside the web application using unsupervised learning approach namely K-medoids and then comes Kar et al [19] with SQLi-GoT a web application firewall base on Support Vector Machine (SVM) clustering classifier to detect SQL Injection. This web application firewall is updated by Pinzon et al that explore two direction first visualization and second is detection, which achieved this direction by a multi agent system called idMAS-SQL with two different classifiers namely Neural Network and SVM. In this research the authors conduct experiment research about concurrent implementation of web application firewall (WAF) and hardening broken authentication (HBA) [20] that able to enhance security in web applications [6], [21].

Methods

There are 3 objects in the experiment methodology in Fig 1., the first object is a web application where security methods and features are implemented, the second is a cyber-attack or penetration testing scenario obtained from OWASP 2017 and the third object is a security point and network access time performance result, the third object will be produced when the second object that is cyber-attack scenario launched on the web application object. The web application object consists of four web applications: 1)

web application without security method and features, 2) web application with WAF, 3) web application with Hardening Broken Authentication, 4) web application with WAF and HBA (the proposed security method). Fig 1. shows the overall experiment that will be conducted by the authors.

Web Application

The web application script is built on Zend Framework (PHP 7.2) and the scripts are stored in a virtual box server, the virtual box server using Ubuntu Server 16.04 LTS and Apache2 web server, the authors created four web applications with a list of security features in Table 2. The first web application is not equipped without any security features, in reason to get an insight of network access time performance. This network access time performance will be shown in section 4 of the experiment results, the second web application is equipped with web application firewall method, the third web application is equipped with the Hardening Broken Authentication method, and the fourth web application is equipped with concurrent implementation of firewall and hardening broken authentication.

Table 2
List Of Web Application Experiment Objects

Number of Web Application	Web Application Method / Feature
1	Web application without any security feature
2	Web application with WAF
3	Web application with Hardening Broken Authentication

Web Application Security Method / Feature

The specification of Web Application Firewall - Hardening Broken Authentication method used in the experiment for Hardening Broken Authentication are session management, brute-force protection and restrict weak passwords and as for web application firewall is placed between the web application and internet by implement the concurrent of WAF-HBA, the web application is more secure to defend cyber-attack scenario such as SQL Injection, DDoS attack, Cross-Site-Scripting and securing user authentication. Hardening Broken Authentication method behavior are:

(1) Session Management – which mean token session changes every time user login and if there's no interaction in web application within 2 days the token login will be changed, to prevent the attackers maintain control on user id in web application.

(2) Brute-Force Protection – the authors create limit on how many wrong passwords input on web application to prevent cyber-attack user trying to login using brute force mechanism.

(3) Restrict weak password – there are three level created to identify the password weakness weak level indicate there are no combination of word, number nor symbol, normal level indicates the password there are combination word and number

with a minimal length of 8 in the password, strong level there are combination word, number and symbol with a minimal length of 8 in the password.

Cyber-attack/Penetration testing scenarios

In Table 1 shows the list of several experiments of cyber-attack scenarios, the list of cyber-attack scenarios is obtained from Open Web Application Security Project 2017 (OWASP) first and second rank scenario, these cyber-attack scenarios will be launched on four web application objects in Table 1. There are two cyber-attack type, the first is injection and the second is broken authentication each cyber-attack type has eight scenario and tools used to launch cyber-attack scenarios, as for "manual" is not required any tool used to launch the cyber-attack scenarios.

Evaluation Methodology

Evaluation Methodology in Table 3 there is list of response codes, response descriptions, and security points. the security point and evaluation methodology are based on Resta study, the authors used points and indicate it with the phrases of more and less secure because several popular studies state there is no absolute security in web application [21], [22]. the most highest point is gained from code R4 and the lowest is gained from code R1, the more higher security point is collected from the Table 3 indicating more secure of web application objects, the lower security point is collected from Table 3 is indicating less secure of web application objects. The evaluation is also complemented with network time access performance evaluation in milliseconds.

Table 3
Response Code Status

Code	Description	Security Point
R1	The Cyber-attack scenario access succeeds, launch succeed, the hacker is able to retrieve/steal the information or retrieve the result. (Not Secured, Danger Status)	1
R2	Status 200 or 500 (Internal Server Error), the cyber-attack scenario successfully accesses however, the web application showed the warning or error messages (Potential of Vulnerability, Warning Status)	2
R3	Status 200 or 404 (Not Found), the cyber-attack scenario successfully accesses, the attacker is not found any critical information (Potential of Vulnerability, Info Status)	3
R4	Server Status 403 (Forbidden) the Cyber-attack scenario is not successfully accessed.	4

Result And Discussion

After the methodology is conducted, the experiment resulted in security score and network time access for each cyber-attack scenario that was launched on four web

application objects. In table 4, table 5 and table 7 there are four web applications and security scoring based on the point collected. The first web application object without any security methods and features suffers vulnerabilities in all cyber-attack scenario: injection type and broken authentication type. the vulnerability is shown due to the acquisition of 1 point for every cyber-attack scenario. This one-point acquisition means the hacker or attacker is successfully launched the cyber-attack and is able to access, retrieve and damage the information. As we can see in Table 6 and 7 the first web application has a 429ms average network access time, it is the fastest access time compared to other web application objects, on the other hand, this object collected the least point in total of 16 security point and 8 point each cyber-attack type.

Table 4
Injection Cyber-Attack Type Experiment Result

Cyber-attack Scenario	HBA	Time	WAF	Time	HBA and WAF	Time
SQL / Blind Injection	2	92ms	4	93ms	4	93ms
XML Injection	3	93ms	4	93ms	4	95ms
Code Injection LFI	1	88ms	4	87ms	4	88ms
Code Injection RFI	1	89ms	1	90ms	1	90ms
CSRF Injection	3	85ms	1	86ms	4	86ms
XSS Injection	3	91ms	4	90ms	4	89ms
Host Header Injection	3	91ms	4	90ms	4	89ms
OS Command Injection	1	88ms	4	87ms	4	88ms

Table 5
Broken Authentication Cyber-Attack Experiment Result

Cyber-attack Scenario	HBA	Time	WAF	Time	HBA and WAF	Time
Injection	17 Point	707ms	26 Point	709ms	**29 Point**	715ms
Broken Auth	20 Point	180ms	14 Point	183ms	**24 Point**	279ms
Total	37 Point	887ms	40 Point	892ms	**53 Point**	994ms
Average	18,5 Point	443,5ms	20 Point	446ms	**26,5 Point**	497ms
Injection	17 Point	707ms	26 Point	709ms	**29 Point**	715ms
Broken Auth	20 Point	180ms	14 Point	183ms	**24Point**	279ms
Total	37 Point	887ms	40 Point	892ms	**53 Point**	994ms
Average	18,5 Point	443,5ms	20 Point	446ms	**26,5 Point**	497ms

Table 6
Overall Score Of The Web Application Security Point

Cyber-attack Scenario	HBA	Time	WAF	Time	HBA and WAF	Time
Password / Information Not Encrypted	1	0	3	0	1	0
Sensitive Data Exposure	1	0	4	0	1	0
Insecure Deserialization	1	87	3	90	1	91
Security Misconfiguration	1	0	1	0	1	0
Using Components with Known Vulnerabilities	1	88	3	90	4	92
Insufficient Logging and Monitoring	1	0	1	0	4	0
Broken Access Control	1	0	4	0	1	0
Password / Information Not Encrypted	1	0	3	0	1	0

Table 7
Score Of The First Web Application Without Any Security Features

Cyber-attack Scenario	Without Security Features	Time (ms)
SQL / Blind Injection	1 Point	87
XML Injection	1 Point	89
Code Injection LFI	1 Point	84
Code Injection RFI	1 Point	86
CSRF Injection	1 Point	83
XSS Injection	1 Point	89
Host Header Injection	1 Point	80
OS Command Injection	1 Point	85
Password / Information Not Encrypted	1 Point	0
Sensitive Data Exposure	1 Point	0
Insecure Deserialization	1 Point	87
Security Misconfiguration	1 Point	0
Using Components with Known Vulnerabilities	1 Point	88
Insufficient Logging and Monitoring	1 Point	0
Broken Access Control	1 Point	0
Injection	**8 Point**	**68**
Broken Auth	**8 Point**	**175**
Total	**16 Point**	**858**
Average	**8 Point**	**429**

The second web application object is implemented with the Hardening Broken Authentication (HBA) method, 3 out of 8 injection types of cyber-attack scenarios were successfully launched with the acquisition of 1 point, the cyber-attack scenario carried out are Code Injection RFI, Code Injection LFI, and OS Command Injection, with HBA methods there is no four-point acquisition for each cyber-attack scenarios. While in Table 5 for the cyber-attack type of broken authentication, HBA methods successfully secure 2 out of 8 cyber-attacks with four-point acquisition namely Access Broken Control and Sensitive Data Exposure. The second web application object has 443.5ms network access time with needed 14ms differentiation compared to the first web application objects. The total score of the second web application in both cyber-attack is 37 points with 20 points on broken authentication cyber-attack type and 17 points on Injection cyber-attack type.

The third web application object is implemented with WAF security features, the third web application is able to secure 2 out of 8 cyber-attack injection types were successfully launched with the methods acquisition point of 1 the cyber-attack namely CSRF Injection and Code Injection RFI. However, the WAF security methods successfully secure 6 out 8 cyber-attack scenarios with an acquisition point of 4 with a total point collected 26 Points it's surprisingly higher than the two web application objects above. As for broken auth cyber-attack type, the WAF method suffers vulnerability with a total collected point is 14 where 6 out 8 cyber-attack scenarios successfully launch with security point acquisition of 1.

The average network access is 2.5ms, but the overall score improves from 37 points to 40 points. the fourth web application is implemented with concurrent use of hardening broken authentication and web application firewall security features, this proposed security method and features are able to improve the total security point collected from 37 to 40 points to 53 points, the total security point of the fourth web application is partly coming from 29 point with cyber-attack injection type and 24 points from cyber-attack broken authentication type see Table 6. These two-point collected are the highest compared to the first, second, and third web applications, where the methods are able to secure 7 out 8 cyber-attack injection types with an acquisition point of 4. As for the broken auth cyber-attack type, the methods are able to secure 6 out of 8 cyber-attacks broken auth type.

Conclusion

The third web application object is implemented with WAF security features, the third web application is able to secure 2 out of 8 cyber-attack injection types were successfully launched with the methods acquisition point of 1 the cyber-attack namely CSRF Injection and Code Injection RFI. However, the WAF security methods successfully secure 6 out 8 cyber-attack scenarios with an acquisition point of 4 with a total point collected 26 Points it's surprisingly higher than the two web application objects above. As for broken auth cyber-attack type, the WAF method suffers vulnerability with a total collected point is 14 where 6 out 8 cyber-attack scenarios

successfully launch with security point acquisition of 1. The average network access is 2.5ms, but the overall score improves from 37 points to 40 points. Current development, the experiment result proves that the combination of web application firewall and hardening broken authentication method are needed as standard to protect web application in backend system and frontend system.

The authors would like to thank you the Bina Nusantara University especially the BINUS Graduate Program – Master of Computer Science Program for helping the authors in the process of learning and explore the cyber security topics and National Police of Republic Indonesia as my workplace to express idea and develop this research.

BIBLIOGRAPHY

Statista.com, "Global digital population as of January 2021," 2021. https://www.statista.com/statistics/617136/digital-population-worldwide/ (accessed Oct. 23, 2021).

Datareportal.com, "Digital 2021: Indonesia," 2021. https://datareportal.com/reports/digital-2021-indonesia (accessed Oct. 23, 2021).

BSSN, "Rekap Serangan Siber," 2020. https://bssn.go.id/rekap-serangan-siber-januari-april-2020/ (accessed Oct. 23, 2021).

L. Sun, H. Zhang, and C. Fang, "Data security governance in the era of big data: status, challenges, and prospects," *Data Sci. Manag.*, vol. 2, pp. 41–44, Jun. 2021, doi: 10.1016/j.dsm.2021.06.001.

D. Wichers and J. Williams, "Owasp top-10 2017," *OWASP Found.*, 2017.

Ibor and J. Obidinnu, "System Hardening Architecture for Safer Access to Critical Business Data," *Niger. J. Technol.*, vol. 34, no. 4, p. 788, Oct. 2015, doi: 10.4314/njt.v34i4.17.

M. Aydos, Ç. Aldan, E. Coşkun, and A. Soydan, "Security testing of web applications: a systematic mapping of the literature," *J. King Saud Univ. Inf. Sci.*, 2021.

F. Ö. Sönmez, "Security qualitative metrics for open web application security project compliance," *Procedia Comput. Sci.*, vol. 151, pp. 998–1003, 2019, doi: https://doi.org/10.1016/j.procs.2019.04.140.

Robinson, M. Akbar, and M. A. Fadhly Ridha, "SQL Injection and Cross Site Scripting Prevention using OWASP ModSecurity Web Application Firewall," *JOIV Int. J. Informatics Vis.*, vol. 2, no. 4, p. 286, Aug. 2018, doi: 10.30630/joiv.2.4.107.

M. A. Calles, "Authentication and Authorization," in *Serverless Security*, Berkeley, CA: Apress, 2020, pp. 229–256.

Sunardi, I. Riadi, and P. Ananda, "Vulnerability Analysis of E-voting Application using Open Web Application Security Project (OWASP) Framework," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, 2019, doi: 10.14569/IJACSA.2019.0101118.

H. Poston, "Mapping the OWASP Top Ten to Blockchain," *Procedia Comput. Sci.*, vol. 177, pp. 613–617, 2020, doi: 10.1016/j.procs.2020.10.087.

J. D'Hoinne, A. Hils, and C. Neiva, "Magic quadrant for web application firewalls," Technical report, Gartner, Stamford, CT, 2014.

Anderies, B. A. Jabar, R. Yunanda, and A. A. S. Gunawan, ."The development of a smart door decision system, based on pir sensor, embedded face recognition and

server request using ttgo esp 32," *ICIC Express Lett. Part B Appl.*, vol. 12, no. 10, pp. 965–970, 2021, doi: 10.24507/icicelb.12.10.965.

N. M. Thang, "Improving efficiency of web application firewall to detect code injection attacks with random forest method and analysis attributes HTTP request," *Program. Comput. Softw.*, vol. 46, no. 5, pp. 351–361, 2020.

Appelt, C. D. Nguyen, and L. Briand, "Behind an application firewall, are we safe from SQL injection attacks?," 2015, doi: 10.1109/ICST.2015.7102581.

Appelt, C. D. Nguyen, A. Panichella, and L. C. Briand, "A Machine-Learning-Driven Evolutionary Approach for Testing Web Application Firewalls," *IEEE Trans. Reliab.*, vol. 67, no. 3, pp. 733–757, 2018, doi: 10.1109/TR.2018.2805763.

M. Ceccato, C. D. Nguyen, D. Appelt, and L. C. Briand, "SOFIA: An automated security oracle for black-box testing of SQL-injection vulnerabilities," in *2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2016, pp. 167–177.

D. Kar, S. Panigrahi, and S. Sundararajan, "SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM," *Comput. Secur.*, vol. 60, pp. 206–225, 2016.

S. K. Choi, C.-H. Yang, and J. Kwak, "System Hardening and Security Monitoring for IoT Devices to Mitigate IoT Security Vulnerabilities and Threats," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 2, pp. 906–918, Feb. 2018, doi: 10.3837/tiis.2018.02.022.

J. Saleem and M. Hammoudeh, "Defense methods against social engineering attacks," in *Computer and network security essentials*, Springer, 2018, pp. 603–618.

H. Cavusoglu, B. Mishra, and S. Raghunathan, "A model for evaluating IT security investments," *Commun. ACM*, vol. 47, no. 7, pp. 87–92, 2004.