Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Website Penetration Analysis Against XSS Attacks using Payload Method

Luthfi Arian Nugraha

Faculty of Science and Technology, Muhammadiyah University of Sidoarjo, Sidoarjo, Indonesia

email: luthfiarian@umsida.ac.id

ARTICLE INFO

Article history:
Received 26 January 2024
Revised 12 June 2024
Accepted 14 June 2024
Available 28 June 2024

Keywords:
XSS
Payload
Penetration testing
Website

IEEE style in citing this article:
L. Arian Nugraha , "Website Penetration Analysis Against XSS Attacks using Payload Method," Journal of Innovation Information Technology and Application (JINITA), vol. 6, no. 1, pp. 37–44, Jun. 2024.

ABSTRACT

This research aims to analyze the effectiveness of various penetration testing methods in identifying and mitigating XSS (Cross-Site Scripting) vulnerabilities in web applications. XSS is a type of web security attack that takes advantage of weaknesses in web applications to insert malicious code into web pages displayed to users. This attack can steal user data, take over user sessions, or spread malware. This research uses a penetration testing method with a black-box approach, where the researcher does not know the construction of the system being tested. Tests were conducted on 10 random websites, including 5 open-source websites and 5 commercial websites. The test results show that the payload method used is effective in exploiting XSS vulnerabilities on some websites. Of the 10 websites tested, 6 of them were successfully exploited using different payload methods. This research highlights the importance of using open-source penetration testing tools in detecting and addressing security vulnerabilities in web applications. These tools are easy to implement, supported by extensive documentation, and have a strong community. This research also emphasizes the importance of a deep understanding of how penetration testing tools work to identify and address security vulnerabilities. To address XSS vulnerabilities, this research recommends good programming techniques such as programming language updates, use of OOP (Object-Oriented Programming), MVC (Model-View-Controller) concepts, and use of frameworks. Further research can be done to develop and test new payload methods, explore the use of other penetration testing tools, and test security vulnerabilities in other types of web applications.

1. INTRODUCTION

The prevalence of vulnerabilities in information technology has rendered numerous websites susceptible to unauthorized individuals seeking to exploit them for unauthorized access or hacking purposes [1]. XSS attacks are a type of web security attack that takes advantage of weaknesses in web applications to insert harmful code into web pages that are shown to users [2]. This malevolent code can subsequently be employed for diverse objectives, including pilfering user data, commandeering user sessions, or disseminating malware [3].

The purpose of this research is to analyze the effectiveness of various penetration testing methods in identifying and mitigating XSS vulnerabilities in web applications. By understanding the impact of XSS attacks and comparing different testing techniques, this study aims to provide practical recommendations for improving web security.

The problem addressed in this research is the increasing prevalence of XSS attacks, which exploit vulnerabilities in web applications to insert malicious code. These attacks pose significant risks to user data,

*) Corresponding Author : luthfiarian@umsida.ac.id

session integrity, and overall web security. XSS attacks can have severe consequences, including the theft of sensitive user data, hijacking of user sessions, and the spread of malware. These impacts highlight the critical need for effective detection and mitigation strategies to protect web applications from such threats.

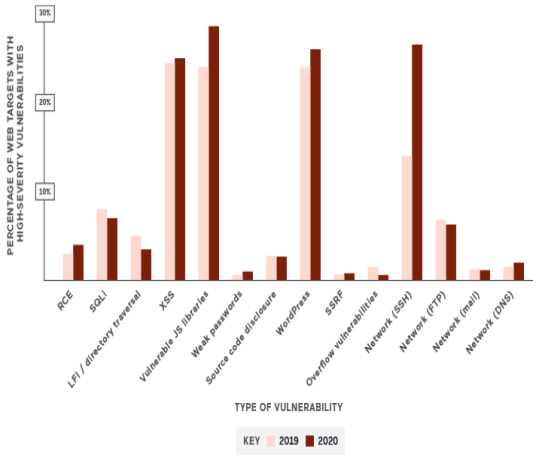

Figure 1. 2021 Acunetix Statistical Report

Acunetix published its 2021 annual report during the spring of 2021, revealing that XSS assaults accounted for 24.5% in 2019 and increased to 25% in 2020. Figure 1 displays a list of 14 vulnerabilities, including XSS assaults in the period of 2019-2020. These attacks experienced a very small increase, with a percentage rise of 0.5%. This proportion represents the highest 4 out of 14 attacks [4].

The previous research from S. Rawat, T. Bhatia, and E. Chopra [5] examined the utilization of penetration test scripts to exploit vulnerabilities in web applications. This publication uses both manual and automated penetration testing techniques to exploit vulnerabilities in web applications. The data presented in this publication demonstrates that penetration test scripts may effectively and expeditiously exploit online application vulnerabilities, such as XSS, SQL Injection, and Cross-Site Request Forgery (CSRF) vulnerabilities. Cybersecurity professionals can utilize penetration test scripts to detect and rectify vulnerabilities in web applications.

E. Chatzoglou, G. Kambourakis, and C. Kolias [6] experimented to evaluate the web management of Wireless Application Protocol (WAP) devices. During the XSS penetration testing, multiple phases were conducted, resulting in the discovery of a significant number of vulnerabilities. Specifically, a total of 29 vulnerabilities were identified.

2. RESEARCH METHODS

The methodology used in this research is a penetration test, which does not constrain the results . In addition to using the penetration test methodology, this research also uses a blackbox approach, where in the test, the researcher is on the outside or does not know the construction of the system [7].

2. 1. Tes flow

Figure 2 is a test flow in collecting some information to get the target to be penetrated [8].

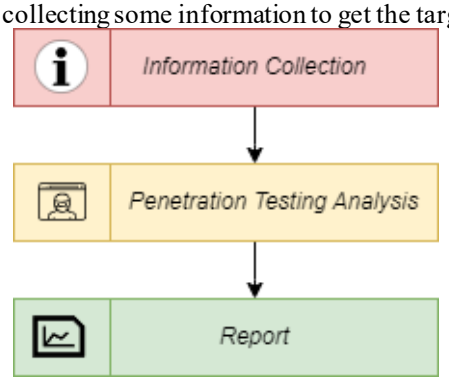

Figure 2. Test stages

a. Information Collection

The collection of information in this case contains the website to be carried out penetration tests and the parameters used as the focus of research [9].

b. Penetration Testing Analysis

Penetration testing analysis is the stage of implementing penetration on the intended platform. This stage generates performance data as a result of penetration reports [10].

c. Report

Reports is the last stage in presenting data that has been processed so that it is easy to understand. This stage can also be interpreted as the presentation of information derived from data that has been collected [11].

2. 2. XSS test pipeline design

XSS is a hacker technique to send payload data to corrupt data presented by a web server [12].

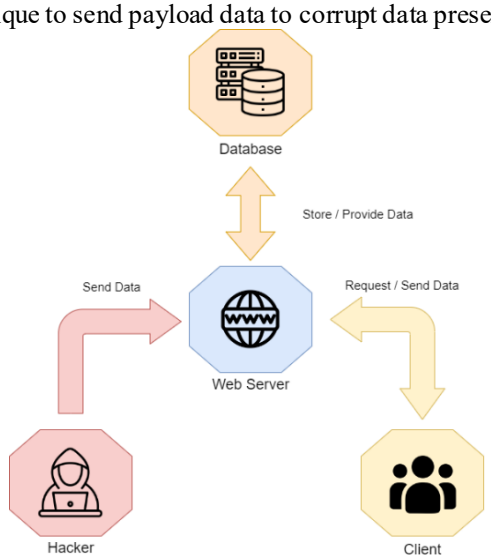

Figure 3. XSS Test Pipeline Design

Researchers ostensibly become hackers to deliver payload data to corrupt the data presented by the web server, this happens when the web application does not filter user input properly. Unfiltered user input can contain malicious code, such as JavaScript scripts [13], then received by the web server and sent to the website's database as shown in Figure 3. Users who request/send data will receive error messages or unwanted data by the user [14].

3. RESULTS AND DISCUSSION

3.1. Information collection

Researchers will simulate attacks on 10 random websites, including 5 open-source websites and 5 commercial websites as shown in Table 1 to find weaknesses or security holes that can be exploited by attackers to perform malicious actions, such as stealing sensitive data, damaging the system, or disrupting website operations.

Table 1. List of websites tested

Type of Website	Name of Website	Method	Target
Public School	Site-A	GET	Search
Private School	Site-B	GET	Search
Community Forum	Site-C	POST	Form
Personal Blog	Site-D	POST	Form
Commercial Blog	Site-E	POST	Login
Entertainment	Site-F	GET	Form
State Service Provider	Site-G	GET	Form
Private Service Provider	Site-H	GET	Form
Freelance	Site-I	GET	Search
Business	Site-J	GET	Form

3.2. Penetration testing analysis

This study used an open-source penetration test tool. This tool was chosen for several important reasons. Firstly, the open-source nature of these tools makes them easy to implement, which is very important in cybersecurity research. Second, the extensive documentation for each tool ensures that the penetration test process can be performed to a consistent and repeatable standard. Finally, the strong community support for these tools provides additional assurance in terms of security updates and quick problem resolution. The following is an analysis of each tool in penetration testing on the 10 websites shown in table 1 [15].

Table 2. Payload method success representation

Name of Website	Payload	Processing Time	Yield Status
Site-A	<DEtAilS%09OnPOIntErEntER+=+(prompt)``%0ty//	265ms	Not
Site-B	<D3V/+/onMoUseOver+=+confirm()>v4fc$	54ms	Not
Site-C	<HTMl/+/onpOINTereNtER%0d=%0da=prompt,a()%0ty//	65ms	Not
Site-D	<d3v%0doNpOINterentEr%0d=%0d(prompt)``%0ty> v4fc$	70ms	Not
Site-E	<a%0dONPOintEReNtER%09=%09(prompt)``> v4fc$	90ms	Yes
Site-F	<A%09OnpOINTeRENteR%09=%09(prompt)``%0ty> v4fc$	2535ms	Yes
Site-G	<a%0dOnmOuseOVer%0d=%0dconfirm()%0ty// v4fc$	3125ms	Yes
Site-H	<d3v%0dOnPOINTerENTer%0a=%0aa=prompt,a()//v4fc$	1276ms	Yes
Site-I	<HTmL/+/OnMOUSEOvER%0a=%0aa=prompt,a()//	7263ms	Yes
Site-J	<dETailS%0aONpoinTErenTER%0a=%0aa=prompt,a()%0ty//	1083ms	Yes

Table 2 shows the execution results of the XSS penetration test. The data does not all show the success of the execution process. However, the successful results show that the payload method used is effective in exploiting XSS vulnerabilities on some websites. This success highlights the importance of using open-source penetration testing tools in detecting and addressing security vulnerabilities.

Figure 4. XSS penetration testing results

Figure 4 shows how the penetration test tool used in this research can generate complex malicious code from a simple line of code. This process illustrates how easy it is for an attacker to insert malicious code into a vulnerable web application. In this example, a single line of code generated by the tool could evolve into seven lines of malicious code that could be used for a variety of malicious purposes, such as stealing user data, taking over a user session, or spreading malware.

The execution of this tool also highlights the importance of a deep understanding of how penetration testing tools work and how they can be used to identify and address security vulnerabilities. Using these tools, researchers can simulate real attacks and test how effective existing security systems are in protecting web applications from such threats.

3.3. Penetration test result solution

a. Programming language updates

The programming language has updated packages to maintain data integrity for each implementation of the application that is created [16]. The programming language has the latest stable version as in Figure 5, Hypertext Preprocessor (PHP) 8.2.12 uploaded on October 26, 2023.

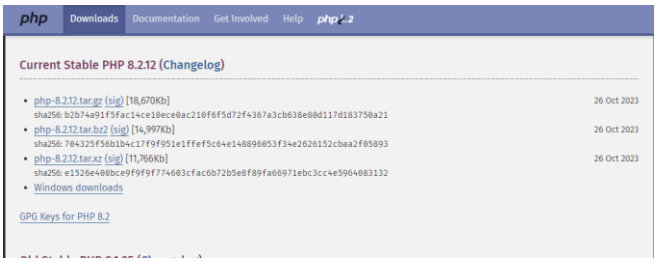

Figure 5. The latest update of the stable PHP package

b. Implementing of object-oriented programming (OOP)

```php
<?php
// Membuat kelas Hewan
class Hewan {
  // Membuat properti nama dan jenis
  public $nama;
  public $jenis;

  // Membuat konstruktor kelas Hewan
  public function __construct($nama, $jenis) {
    // Menginisialisasi properti dengan nilai yang diberikan
    $this->nama = $nama;
    $this->jenis = $jenis;
  }

  // Membuat metode untuk menampilkan informasi hewan
  public function info() {
    echo "Hewan ini bernama $this->nama dan termasuk jenis $this->jenis.\n";
  }
}

// Membuat objek hewan1 dari kelas Hewan
$hewan1 = new Hewan("Kucing", "Mamalia");

// Memanggil metode info pada objek hewan1
$hewan1->info();
?>
```

Figure 6. Simple OOP PHP code script

OOP is one of the important functions in building an application because OOP has a feature called encapsulation. Encapsulation is a feature to hide or withhold information that is not needed by the program [17]. Figure 5 example of a PHP programming language code script with OOP implementation.

c. Model-View-Controller (MVC) implementation

XSS can be prevented using the MVC application concept method because MVC is implemented separately from the User-Interface (UI) display and program logic [18]. MVC has the flow diagram in Figure 7.

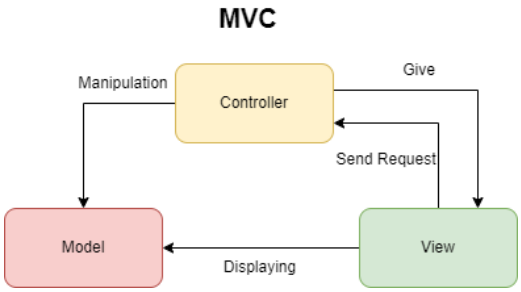

Figure 7. Flow chart MVC

d. Framework implementation

A framework of a collection of code that has been written by other developers and has been tested for security, so it is very minimal against XSS attacks [19]. Laravel, Symfony, and CodeIgniter are examples of frameworks for PHP programming languages such as Figure 8.

Figure 8. Example frameworks from the PHP programming language

e. Rest API implementation

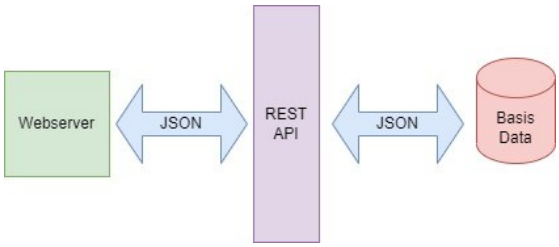

Figure 9. How the rest of API works

Representational Application Programming Interface (Rest API) is a bridge between the web server and the database. This tool is to minimize the presence of irresponsible people accessing the database which must be maintained integrity [20]. Figure 9 explains that the web application has 3 engines with interoperability, namely how to communicate, send, and receive formatted files: JavaScript Object Notation (JSON) as an interpretation of the required data [21].

3.4. Discussion of penetration test results

This research describes the results of penetration testing conducted on 10 websites, consisting of 5 open-source websites and 5 commercial websites. These tests aim to find weaknesses or security holes that can be exploited by attackers to perform malicious actions, such as stealing sensitive data, damaging systems, or disrupting website operations. The test results show that the payload method used is effective in exploiting XSS vulnerabilities on some websites. Of the 10 websites tested, 6 of them were successfully exploited using different payload methods. This shows that XSS is one of the most common web security attacks and can be used to steal user data, take over user sessions, or spread malware.

This research also highlights the importance of using open-source penetration testing tools in detecting and addressing security vulnerabilities in web applications. These tools are not only easy to implement, but are also supported by extensive documentation and a strong community, ensuring that the penetration testing process can be performed to a consistent and repeatable standard. In addition, this research emphasizes the importance of a deep understanding of how penetration testing tools work to identify and address security vulnerabilities. Using these tools, researchers can simulate real attacks and test how effective existing security systems are in protecting web applications from such threats.

To address XSS vulnerabilities, this research recommends some good programming techniques, such as programming language updates, use of OOP (Object-Oriented Programming), MVC (Model-View-Controller) concepts, and use of frameworks. These techniques can help prevent XSS attacks by ensuring that user input is properly filtered and data sent to the web server is safe from malicious code. Further research that can be done includes developing and testing new payload methods to improve the effectiveness of XSS vulnerability detection and exploitation, exploring the use of other penetration testing tools and comparing their effectiveness with the tools used in this study, conducting further research on more effective XSS attack mitigation and prevention techniques, as well as testing security vulnerabilities in other types of web applications and developing more specific solutions for each type of application.

3.5. Comparison with previous research

Previous research by S. Rawat, T. Bhatia, and E. Chopra as well as E. Chatzoglou, G. Kambourakis, and C. Kolias focused on using penetration test scripts and XSS penetration testing to exploit vulnerabilities in web applications and WAP devices.

This research goes on to use blackbox penetration test methods on different types of websites and emphasizes the importance of using open-source penetration test tools as well as good programming techniques to prevent XSS attacks. This research extends the scope by testing more websites and provides practical recommendations to improve the security of web applications from XSS attacks.

4. CONCLUSION

The results of this study show that the payload methods used are effective in exploiting XSS vulnerabilities on several websites. Out of the 10 websites tested, 6 of them were successfully exploited using different payload methods. The findings of this research are that XSS (Cross-Site Scripting) is one of the common web security attacks and can be used to steal user data, take over user sessions, or spread malware. The use of open-source penetration testing tools is very effective in detecting and addressing security vulnerabilities in web applications. An in-depth understanding of how penetration testing tools work is essential for identifying and addressing security vulnerabilities. Implementation of good programming techniques such as programming language updates, use of OOP (Object-Oriented Programming), MVC (Model-View-Controller) concepts, and use of frameworks can help prevent XSS attacks. Further research that can be done includes developing and testing new payload methods to improve the effectiveness of XSS vulnerability detection and exploitation, exploring the use of other penetration testing tools and comparing their effectiveness with the tools used in this study, conducting further research on more effective XSS attack mitigation and prevention techniques, as well as testing security vulnerabilities in other types of web applications and developing more specific solutions for each type of application.

REFERENCES

[1] G. E. Rodríguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-site scripting (XSS) attacks and mitigation: A survey," *Comput. Netw.*, vol. 166, p. 106960, Jan. 2020, doi: 10.1016/j.comnet.2019.106960.

[2] V. S. Stency and N. Mohanasundaram, "A Study on XSS Attacks: Intelligent Detection Methods," *J. Phys. Conf. Ser.*, vol. 1767, no. 1, p. 012047, Feb. 2021, doi: 10.1088/1742-6596/1767/1/012047.

[3] S. Kumar, S. Pathak, and J. Singh, "An enhanced digital forensic investigation framework for XSS attack," *J. Discrete Math. Sci. Cryptogr.*, vol. 25, no. 4, pp. 1009–1018, May 2022, doi: 10.1080/09720529.2022.2072424.

[4] "The Invicti AppSec Indicator Spring 2021 Edition: Acunetix Web Vulnerability Report," Acunetix . Accessed: Nov. 21, 2023. [Online]. Available: https://www.acunetix.com/white-papers/acunetix-web-application-vulnerability-report-2021/

[5] S. Rawat, T. Bhatia, and E. Chopra, "Web Application Vulnerability Exploitation using Penetration Testing scripts," *Int. J. Sci. Res.*, vol. 6, no. 1, 2020.

[6] E. Chatzoglou, G. Kambourakis, and C. Kolias, "Your WAP Is at Risk: A Vulnerability Analysis on Wireless Access Point Web-Based Management Interfaces," *Secur. Commun. Netw.*, vol. 2022, pp. 1–24, Feb. 2022, doi: 10.1155/2022/1833062.

[7] F. Prasetyo, U. R. Jannah, and M. U. Mansyur, "Penggunaan Stb Sebagai Media E-Learning Berbasis Moodle," JURNAL INFORMATIKA, vol. 23, no. 01, 2023, doi: 10.30873/ji.v23i1.3523

[8] M. Hasibuan and A. M. Elhanafi, "Penetration Testing Sistem Jaringan Komputer Menggunakan Kali Linux untuk Mengetahui Kerentanan Keamanan Server dengan Metode Black Box: Studi Kasus Web Server Diva Karaoke.co.id," *Sudo J. Tek. Inform.*, vol. 1, no. 4, pp. 171–177, Dec. 2022, doi: 10.56211/sudo.v1i4.160.

[9] C. B. Setiawan, D. Hariyadi, A. Sholeh, and A. Wisnuaji, "Pengembangan Aplikasi Information Gathering Berbasis HybridApps," *INTEK J. Inform. Dan Teknol. Inf.*, vol. 5, no. 1, Art. no. 1, May 2022, doi: 10.37729/intek.v5i1.1729.

[10] Y. A. Pohan, Y. Yuhandri, and S. Sumijan, "Meningkatkan Keamanan Webserver Aplikasi Pelaporan Pajak Daerah Menggunakan Metode Penetration Testing Execution Standar," *J. Sistim Inf. Dan Teknol.*, pp. 1–6, Sep. 2021, doi: 10.37034/jsisfotek.v3i1.36.

[11] F. Y. Fauzan and S. Syukhri, "Analisis Metode Web Security PTES (Penetration Testing Execution And Standart) Pada Aplikasi E-Learning Universitas Negeri Padang," *Voteteknika Vocat. Tek. Elektron. Dan Inform.*, vol. 9, no. 2, Art. no. 2, Jun. 2021.

[12] B. B. Gupta, P. Chaudhary, and S. Gupta, "Designing a XSS Defensive Framework for Web Servers Deployed in the Existing Smart City Infrastructure:," *J. Organ. End User Comput.*, vol. 32, no. 4, pp. 85–111, Oct. 2020, doi: 10.4018/JOEUC.2020100105.

[13] C. Lv, L. Zhang, F. Zeng, and J. Zhang, "Adaptive Random Testing for XSS Vulnerability," in *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, Dec. 2019, pp. 63–69. doi: 10.1109/APSEC48747.2019.00018.

[14] P. Chaudhary, B. B. Gupta, X. Chang, N. Nedjah, and K. T. Chui, "Enhancing big data security through integrating XSS scanner into fog nodes for SMEs gain," *Technol. Forecast. Soc. Change*, vol. 168, p. 120754, Jul. 2021, doi: 10.1016/j.techfore.2021.120754.

[15] J. R. Dora and K. Nemoga, "Ontology for Cross-Site-Scripting (XSS) Attack in Cybersecurity," *J. Cybersecurity Priv.*, vol. 1, no. 2, Art. no. 2, Jun. 2021, doi: 10.3390/jcp1020018.

[16] N. P. Dewi and I. Listiowarni, "Implementasi Game Based Learning pada Pembelajaran Bahasa Inggris," *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 3, no. 2, pp. 124–130, Aug. 2019, doi: 10.29207/resti.v3i2.885.

[17] D. P. Y. Ardiana and L. H. Loekito, "Gamification design to improve student motivation on learning object-oriented programming," *J. Phys. Conf. Ser.*, vol. 1516, no. 1, p. 012041, Apr. 2020, doi: 10.1088/1742-6596/1516/1/012041.

[18] E. Bautista-Villegas, "Metodologías agiles XP y Scrum, empleadas para el desarrollo de páginas web, bajo MVC, con lenguaje PHP y framework Laravel," *Rev. Amaz. Digit.*, vol. 1, no. 1, Art. no. 1, Jan. 2022, doi: 10.55873/rad.v1i1.168.

[19] S. Suroto and A. Asman, "Ancaman Terhadap Keamanan Informasi Oleh Serangan Cross-Site Scripting (Xss) Dan Metode Pencegahannya," ZONA KOMPUTER: Program Studi Sistem Informasi Universitas Batam, vol. 11, no. 1, pp. 11-19, 2021.
[20] M. Iqbal and N. Nurwati, "Penerapan Sistem Terintegrasi Menggunakan Restful Api Pada Dealer Management System Panca Niaga Sei Piring," *J. Sci. Soc. Res.*, vol. 6, no. 1, Art. no. 1, Feb. 2023, doi: 10.54314/jssr.v6i1.1161.
[21] C.-O. Truică, E.-S. Apostol, J. Darmont, and T. B. Pedersen, "The Forgotten Document-Oriented Database Management Systems: An Overview and Benchmark of Native XML DODBMSes in Comparison with JSON DODBMSes," *Big Data Res.*, vol. 25, p. 100205, Jul. 2021, doi: 10.1016/j.bdr.2021.100205.