A review of attacks, objects, and mitigations on web services

Rafif Masrur Rauf, Tubagus Eiffel Rivaldo, and Amiruddin Amiruddin *
Sekolah Tinggi Sandi Negara, Bogor, Indonesia

* amir@stsn-nci.ac.id

Abstract. Web service is a technology that continues to develop until now. Web services are needed for the exchange or dissemination of information between applications through a computer network. Various methods have been developed to improve the performance and security of web services. However, the more the increasing performance of web services, the higher the level of vulnerabilities and threats. Some attacks often target on web services are Denial of Service (DoS), Structure Query Language (SQL) injection, and eXtensible Markup Language (XML) injection. There are also many methods or techniques that researchers proposed as mitigations for these attacks. In this research, a literature review has been carried out regarding attacks as well as objects and mitigation of such attacks on web services.

1. Introduction
Web services are needed for the exchange or dissemination of information between applications through a computer network. Web service is a development of web-based applications. The benefit of web services that can easily connect multiple applications makes it widely used and utilized as an industry requirement. The increasing use of web service technology by several organizations raises security vulnerabilities and threats that must be handled properly, otherwise those can have a negative impact and harm the organizations or individuals. One of the security threats of using this web services technology is the increasing number of cyber attacks by exploiting existing vulnerabilities. This problem can become a serious global problem because web services can be accessed by anyone. The most common attacks on web services are XML Injection, XPath Injection, SQL Injection, Spoofing, Denial of Service (DOS), and Man in The Middle Attack [10]. These attacks have not shown signs of diminishing and have instead increased [3]. Creating a secure web service requires knowledge of various attacks and mitigation methods. In this study, a literature review has been conducted focusing on web service attacks and security mitigations.

2. Method and materials

2.1. Review preparation
The first step in conducting this literature review was to develop research questions to be answered. Research questions were made to guide the process of finding and extracting data. In this study, we defined the research questions as the following:
RQ1: What attacks can occur on web services?
RQ2: What mitigation methods are used to mitigate these attacks?
RQ3: What objects of web service are affected by these attacks?

2.2. Method
The next step after developing the research questions was looking up and filtering references.
The search of reference was done by using a combination of key words as follows: attack, web service, vulnerability, security techniques, and security. After the references were collected, data extraction and processing were carried out. Data in the form of attacks, objects of attacks, and mitigation techniques of web service were extracted from references and tabulated.

3. Results and discussion

3.1. Attacks on web services

Several attacks on web services found in the literature are as summarized in Table 1 and a brief description of each attack is given in the following:

a. **SQL Injection.** SQL Injection tries to manipulate application database using query statements like SELECT and UNION. An attacker can execute unwanted commands in the database using this attack. Attackers make indications of errors in the system and then find these errors to be exploited further.

b. **XML Injection.** This attack occurs when the web service fails to validate dangerous XML content. This vulnerability can change the working logic of the system.

c. **Cross Site Scripting (XSS).** XSS is caused by a program or application failure to validate user input before being given back to the client. Attackers can make the server send malicious scripts or HTML to the victim's browser.

d. **XML Signature Wrapping.** XML Signature Wrapping allows attackers to change messages that are signed arbitrarily.

e. **Buffer Overflow.** This attack provides a larger amount of data beyond capacity into the program variable and by doing so arbitrary code can be executed using privileges (root) access. Even worse if the program enters data into the buffer and does not check its size, it will result in buffer overflow.

f. **Spoofing.** Attackers falsify a vector to deceive the recipient that what is being sent is trusted and then gain valid access allowing them to modify or delete sensitive data and control the victim's device.

 g. **Network Eavesdropping.** This happens when an attacker gets access to certain networks. Attackers can capture traffic to get the user's name and password.

h. **Dictionary Attack.** The attacker systematically tests all possible passwords using a dictionary of prepared words. Doing a dictionary attack means trying every word in the dictionary that has been previously prepared until a suitable password is found.

i. **Data Tampering.** Data damage occurs when an attacker alters or modifies legitimate data through the network. Any data sent from the client side can be manipulated by an attacker.

j. **Denial of Service (DoS).** The purpose of this attack varies, one of which deactivates the user's device or network, and prevent the service providers to provide its services.

k. **Man in the Middle Attack.** The attacker cuts the communication link between two parties and communicates to one party without the parties knowing it. Attackers are free to modify the message content and are free to send it.

l. **Oversized payloads.** Sending file with a huge or indefinite size can create and run XDoS attacks.

m. **Web Services Definition Language (WSDL) Scanning.** The WSDL document contains information that explains how to use parameters and information about the details of the method. Scanning into WSDL documents can reveal sensitive information such as message types, operations, port types, bindings, and other methods.

n. **XPath Injection.** An attacker forms a query such as SQL in an XML document using XPath to extract an XML database.
### Table 1. Review result: attacks and mitigations on web service

| No | Attacks                                | Mitigations                                                                                          |
|----|----------------------------------------|-------------------------------------------------------------------------------------------------------|
| 1  | SQL Injection [2][3][4][5][7][6][8]    | Strict SDLC [3], Penetration Testing dengan WS-Attacker [4], input validation [5], CIVS-WS (Command Injection Vulnerability Scanner for Web Services) [8], Hybrid Method [8], HP WebInspect [8], IBM Rational AppScan [8], Acunetix Web Vulnerability Scanner [8], Penetration testing tool [8], Secure Programming [8] |
| 2  | XML Injection [4][6][8]                | Penetration Testing with WS-Attacker [4], Pluggable API as well as security services in the middleware [6], SOAP message tree verification [6], OGSA-P2P framework [8] |
| 3  | XSS (Cross Site Scripting) [3][4][5][8] | Strict SDLC [3], Penetration Testing with WS-Attacker [4], Input validation [5], Hybrid Method [2] |
| 4  | DOS (Denial of Service) [3][4][5][6][8] | Penetration Testing with WS-Attacker [4], Vector Quantization based Intrusion detection system [6], gateway system based on schema hardening to filtering malicious SOAP message [6], XML Auth based step system called CheckWay Gateway [8], Client-Puzzle method [8] |
| 5  | Hijacking [3][4][7]                    | Penetration Testing with WS-Attacker [4]                                                              |
| 6  | Data Tampering [5]                     | Input validation [5]                                                                                |
| 7  | Unauthorized Access [3]                | XML Signatures [4]                                                                                  |
| 8  | Phising [3]                            | XML Signatures [4]                                                                                  |
| 9  | Malware [3]                            | Dynamic Analysis [6], Static Analysis [6]                                                            |
| 10 | Brute Force Attack [3]                 | XML Encryption [4]                                                                                  |
| 11 | Dictionary Attack [5]                  | Input validation [5], cryptography [5], digital certificate [5], digital signature [5], SSL [5][7] |
| 12 | DNS Hijacking [3]                      | XML Signature [4]                                                                                    |
| 13 | XML Signature Wapping Attack [4]       | Penetration Testing with WS-Attacker [4]                                                              |
| 14 | Oversize payload [4][5]                | Penetration Testing with WS-Attacker [4], Parser checking for abnormal condition [5]                  |
| 15 | Spoofing [4][5][6]                     | Using Frameworks Web Service : Apache Axis2, JBossWS native, JBossWS CXF dan .Net Web Services [4], end point reference verification (White List) [4], penetration Testing with WS-attacker [4], input validation [5], cryptography [5], digital certificate [5], digital signature [5], SSL [5][7], automated pluggable API Model for network level [6] |
| 16 | WSDL Scanning [4][5]                   | Penetration Testing with WS-Attacker [4]                                                              |
| 17 | Attack Obfuscation [4]                 | Penetration Testing with WS-Attacker [4]                                                              |
| 18 | BPEL State Deviation [4]               | Penetration Testing with WS-Attacker [4]                                                              |
| 19 | Indirect Flooding [4]                  | End point reference verification (White List)[4]                                                      |
| 20 | Instantiation Flooding [4]             | End point reference verification (White List)[4]                                                      |
| 21 | Attack on XML Encryption [4][7]        | End point reference verification [4]                                                                 |
| 22 | Coercive Parsing [4]                   | Penetration Testing dengan WS-Attacker [4]                                                            |
| 23 | Buffer Overflows [4][5]                | Penetration Testing dengan WS-Attacker [4], Input validation [5]                                    |
| 24 | Network Eavesdropping [5][7]           | Dynamic Analysis [6], Static Analysis [6]                                                            |
| 25 | X-Path Injection [5][6][8]             | A run-time monitoring mechanism [6], tool bernama CIVS-WS (Command Injection Vulnerability Scanner for Web Services) [8], HP WebInspect [8], IBM Rational AppScan [8], Acunetix Web Vulnerability Scanner [8] |
| 26 | Man in the Middle Attack [5]           | Cryptography [5]                                                                                    |
As seen on Figure 1, the type of attacks that most frequently targeting on web services are SQL Injection (13.2%), Denial of Service (DoS) (9.4%), Cross Site Scripting (XSS) (9.4%), Hijacking (5.7%), Spoofing (5.7%), X-Path Injection (5.7%), XML Injection (5.7%). This means that the most common attack on web services is SQL injection. As it is known that SQL injection exploits weaknesses in the program code that does not validate the user input, this indicates that there are still many web service developers who have not given more attention in web service design, especially in validating user input.

![Figure 1. Type of attacks and occurrence frequency](image)

3.2. Attacks mitigation on web services

Several types of attacks describe in the previous section can be approached through the mitigation as summarized in Table 1 and described as following:

a. **Strict SDLC.** Software Development Life Cycle (SDLC) is an application development method that implements strict security on the initial development process, requirement analysis, design, coding, testing, and implementation. Safeguarding the web when the process of building and coding is an effective method that can be used to reduce vulnerabilities in web applications.

b. **XML Encryption.** A standard use of XML by W3C that is used to maintain the confidentiality of XML data.

c. **XML Signature.** A standard for using XML by W3c that is used to guarantee the integrity and authentication of XML data.

d. **Penetration Testing with WS-Attacker.** WS-Attacker tools can be used to perform penetration testing of web services to analyze specific web service attacks and there are no tools that can scan for those specific attacks.

e. **Input Validation.** The process of checking input is performed by the client to ensure that the input does not threaten the system.
f. **Hybrid Method.** This is a combination of static and dynamic analysis. In static analysis, attention will be given during the coding whereas in dynamic analysis a measure can be taken after the application is published or in execution time.

g. **CIVS-WS.** This is a tool developed to detect attacks on web services aiming to detect similar attacks by teaching the system about the types of queries.

h. **OGSA-P2P framework.** A framework for developing peer-to-peer architecture to prevent vulnerabilities in web services in the architecture section.

i. **CheckWay Gateway.** A step-by-step XML-based authentication system developed to prevent DoS attacks. This is installed on the system and a gateway is created between the web service server and the client. Simple Object Access Protocol (SOAP) request messages from client to server must pass this gateway authentication.

j. **ClientPuzzle.** DOS attacks are divided into flood and semantic attacks. To prevent these type of attacks, the ClientPuzzle method was developed. The server asks the client to solve several puzzles from moderate to difficult level. The goal is to make a difference between the person who makes the game and the person who becomes a bot.

k. **Dynamic Analysis.** Dynamic analysis techniques focus on identifying the consistency of output with the input given in runtime.

l. **Static Analysis.** Static analysis is related to code and does not focus on execution. Differences are identified in the code to relate to a technique that is to analyze the vulnerability of the code. This analysis can be done during the development phase.

![Figure 2. Objects of attacks and occurrence frequency](image)

Figure 2 shows that objects in the web service that are most targeted by attackers are XML (25.8%), network (16.1%), SOAP (12.9%), WS addressing (9.7%), back-end apps (6.5%), and session (6.5%). This indicates that XML is the object of the most vulnerable web services and should be the main concern of developers.

### 4. Conclusion

In this study, we have reviewed several literature focusing on attacks, object of attacks, and mitigation techniques on web services. We found that the most common attack on web services
is SQL injection, which indicates that there are still many web service developers who have not given more attention in designing the web services, especially in validating the user input. We also found that the most targeted object on web service is XML and this should be the main concern of developers in developing secure web service applications.

5. Reference

[1] Boncella, R.J., “Web Services and Web Services Security”, Communications of the Association for Information Systems, Volume 14, 2004, pp. 344-363
[2] Faker, Salem A.; Muslim, Mohamed A.; and Dachlan, Harry S., “Systematic Literature Review on SQL Injection Attacks Techniques and Common Exploited Vulnerabilities”, International Journal of Computer Engineering and Information Technology, Vol. 9, N0. 12, December 2017, pp. 284–291
[3] Kaur, Daljit and Kaur, Parminder Dr., “Empirical Analysis of Web Attacks”, Procedia Computer Science, Volume 78, 2016, pp. 298-306
[4] Mainka, C.; Somorovsky, J.; & Schwenk, J., “Penetration Testing Tool for Web Services Security”, Proceedings - 2012 IEEE 8th World Congress on Services, 2012
[5] Moradian, E. & Hakansson, A., “Possible Attacks on XML Web Services”, International Journal of Computer Science and Network Security (IJCSNS), Vol.6 No.1B, January 2006, pp.154-170
[6] Mouli, V. R., & Jevitha, K., “Web Services Attacks and Security - A Systematic Literatur Review”, Procedia Computer Science, Elsevier, Volume 93, 2014, pp.870-877
[7] Kuyoro Shade O; Ibikunle Frank; A Wodele O; and Okolie Samuel O., “Security Issues in Web Services”, International Journal of Computer Science and Network Security (IJCSNS), VOL.12 No.1, 2012, pp.23-27
[8] Sabaz, F., & Celik, Y., “Systematic Literature Review on Security Vulnerabilities and Attack Methods in Web Services”, International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES’18), pp.821-825, 2018
[9] Salas, Marcelo Invert Palma and Martins, Eliane; “Security Testing Methodology for Vulnerabilities Detection of XSS in Web Services and WS-Security”, Electronic Notes in Theoretical Computer Science, Volume 302, 25 February 2014, pp.133-154
[10] Yu, F., & Tung, Y.-Y., “Patcher: An Online Service for Detecting, Viewing and Patching Web”, 47th Hawaii International Conference on System Sciences, pp.4878-4886, 2014