Attack Taxonomy Methodology Applied to Web Services

ARTICLE HISTORY
Received 05 February 2023
Accepted 17 May 2023
Published 08 January 2024

Marcelo I. P. Salas
UNICAMP, University of Campinas
Campinas, SP, Brazil
marcelopalma@ic.unicamp.br
ORCID: 0000-0001-6821-0002

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024

Attack Taxonomy Methodology Applied to Web Services

Marcelo I. P. Salas
UNICAMP, University of Campinas
Campinas, SP, Brazil
marcelopalma@ic.unicamp.br

***Abstract*—** With the rapid evolution of attack techniques and attacker targets, companies and researchers question the applicability and effectiveness of security taxonomies. Although the attack taxonomies allow us to propose a classification scheme, they are easily rendered useless by the generation of new attacks. Web services, owing to their distributed and open nature, present novel security challenges. The purpose of this study is to apply a methodology for categorizing and updating attacks prior to the continuous creation and evolution of new attack schemes on web services. Also, in this research, we collected thirty-three (33) types of attacks classified into five (5) categories, such as brute force, spoofing, flooding, denial-of-services, and injection attacks, in order to obtain the state of the art of vulnerabilities against web services. Finally, the attack taxonomy is applied to a web service, modeling through attack trees. The use of this methodology allows us to prevent future attacks applied to many technologies, not only web services.

***Keywords*—** *Attack taxonomy methodology, web services, brute force, spoofing, flooding, denial-of-services, injection*

***Resumen*—**Con la rápida evolución de las técnicas de ataque y los objetivos de los atacantes, las empresas y los investigadores cuestionan la aplicabilidad y eficacia de las taxonomías de seguridad. Si bien las taxonomías de ataque nos permiten proponer un esquema de clasificación, son fácilmente inutilizadas por la generación de nuevos ataques. Los servicios web, debido a su naturaleza distribuida y abierta, presentan nuevos desafíos de seguridad. El propósito de este estudio es aplicar una metodología para categorizar y actualizar ataques previos a la continua creación y evolución de nuevos esquemas de ataque a servicios web. Asimismo, en esta investigación recolectamos treinta y tres (33) tipos de ataques clasificados en cinco (5) categorías, tales como fuerza bruta, suplantación de identidad, inundación, denegación de servicios y ataques de inyección, con el fin de obtener el estado del arte de las vulnerabilidades contra servicios web. Finalmente, se aplica la taxonomía de ataque a un servicio web, modelado a través de árboles de ataque. El uso de esta metodología nos permite prevenir futuros ataques aplicados a muchas tecnologías, no solo a servicios web.

***Palabras clave*—** *Metodología de taxonomía de ataque, servicios web, fuerza bruta, suplantación de identidad, inundación, denegación de servicios, inyección*

I. Introduction

Taxonomy is the practice and science of categorization and classification of something [1]. To understand the relationship between attacks and defenses in attack taxonomy, the present research proposes a methodology to classify attacks, vulnerabilities, and faults in relation to their features. The taxonomy proposed is constructed to build a better understanding of each attack and present possible countermeasures applied to a technology in constant development, such as web services.

Web Services are modular software applications that can be described, published, located, and invoked over a network such as the World Wide Web [2]. They are also more susceptible to security risks due to their distributed and open nature, although they provide greater connectivity, flexibility, and interoperability as one of the main benefits of this technology.

A web service [3] refers to a comprehensive set of open protocols and standards designed for seamless data exchange between various applications or systems. This versatile technology, implemented in multiple programming languages and compatible with diverse platforms, can use web services to exchange data over computer networks like the Internet in a manner like inter-process communication on a single computer. This interoperability (e.g., between Java and Python or Windows and Linux applications) is due to the use of open standards, and these features make them more attractive while generating new challenges for maintaining information security.

A study quoted in [4] described that the use of web service technology reopened 70% of the vulnerabilities filtered by firewalls. Also, in addition to traditional vulnerabilities, there are new ones in web services that must be considered. According to the OWASP[1] Top 10 [5] and the CWE[2] [6], injection and denial of service attacks were among the most exploited in 2021.

The Simple Object Access Protocol (SOAP), as shown in Fig. 1, used to exchange messages among participants, does not address security by itself, and it can bypass a firewall through the Extensible Markup Language (XML), usually via HTTP [7]. The receiving system interprets the message and sends back a response in the form of another SOAP message under a Service Oriented Architecture (SOA). This vulnerability allows attackers to easily exploit and manipulate the messages to their advantage.

Some research [5], [6], [8], [9] analyzed the web service attacks as a set of threats, which can be mitigated by current security specifications, such as WS-Security, XML-Encryption, XML-Signature, among others. The problem resides in having a partial view of the attack and difficulty classifying it, selecting characteristics of system security based on a taxonomy tested to create a testing plan for web services, and using appropriate countermeasures.

[1] The Open Web Application Security Project (OWASP) is a nonprofit foundation dedicated to improving software security.

[2] Common Weakness Enumeration (CWE) is a universal online dictionary of weaknesses that have been found in computer software.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

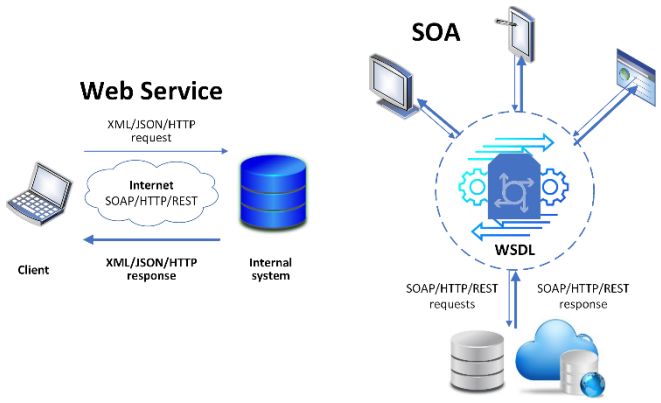

Fig. 1. Illustrates the process of client invokes a web service by sending an XML request services, which then sends back an XML response. It uses many standards such as WSDL3 and SOAP.

This proposal designs and applies a methodology to classify thirty-three (33) web services attacks into five (5) categories, describing their security properties affected, the level at which they develop, and countermeasures, among others, in the section III. For ease of understanding, attack trees are applied to model vulnerabilities against web services.

The rest of the paper is organized as follows. Section II provides a review of related work and existing attack taxonomies. Section III proposes a methodology for attack taxonomy delves into security challenges and attacks on web services with possible countermeasures. Section IV applies the methodology of attack taxonomy. Section V concludes the research, describing the contributions and future work.

II. Related Work

Taxonomy [10] is described as the study of the common principle of scientific classification and includes basis, principles, procedures, and rules. It can be used to indicate the actual categorization of objects, e.g. attack taxonomy that describes vulnerabilities like injection or brute force attack. This section provides an overview of the properties of a taxonomy as well as existing works.

A. Analysis of the Properties of the Taxonomy

According to [11], it is important to define the properties and requirements for a correct classification process in an attack taxonomy, defined as follows:

- Acceptable by the security community.
- Comprehensible taxonomy could be understood by security experts.
- Completeness and exhaustiveness ensure that all types of attacks are covered, and if new ones emerge, the taxonomy can be expanded to include them.
- Determinism is the procedure by which classification occurs and is clearly defined.
- Mutually exclusive means that each attack can only be categorized, at most, into one category.
- Repeatable means that the classification of an attack should be reproducible.
- Constant and defined security terminology means that it makes use of standard and well-established nomenclature in the area.
- Well-defined terms allow a categorization of attacks through precise features.
- Unambiguous means that the taxonomy must have clearly defined classes.
- Usefulness is a requirement that can currently be tested through security testing.

B. A Brief Survey of Attack Taxonomies

Numerous attack taxonomies have been developed over the years to protect web applications and services. Below, we describe these taxonomies based on their potential and relationship with this research.

Chan et al. [14] presented a taxonomy that offers a comprehensive framework for understanding attacks within a new classification. The taxonomy organizes and classifies attacks based on three key parameters: the web services layer, attack methodology, and impact. This proposed taxonomy provides the necessary flexibility to classify emerging web service attacks within a Service-Oriented Architecture (SOA) environment.

Karumanchi and Squicciarini [7] went beyond addressing commonly known web-based vulnerabilities like SQL injection and session replay. They also conducted an examination of web service-specific vulnerabilities, highlighting the potential for attacks arising from subpar service construction and inadequate maintenance. In their comprehensive analysis, they classified each vulnerability based on a novel taxonomy, discussing potential solutions and associated impacts. Additionally, they proposed real-time analysis-based detection methods. However, it should be noted that their taxonomy does not include the essential tools for the classification of attacks.

Derbyshire et al. [1] applied two approaches to evaluate seven taxonomies. The first approach analyzed the criteria used for the taxonomy creation and critical components. The second applied historical attack data to each taxonomy under review, more specifically, attacks in which industrial control systems have been targeted. This combination of methodologies enables a comprehensive exploration of existing taxonomies, offering insights from both theoretical and practical perspectives, thus fostering a deeper understanding of the subject matter.

Panchal et al. [12] introduced an Industrial Internet of Things (IoT) attack taxonomy as a valuable resource for mitigating attack risks. Their research incorporated four key dimensions: attack vector, attack target, attack impact, and attack consequence. These dimensions provide a comprehensive framework for modeling attacks on industrial infrastructures and offer insights into the attack methodology, affected components, and the attacker's objectives. Nevertheless, the taxonomy did not include a breakdown of attacks into specific features, nor did it facilitate reproducibility of the attacks.

Simmons et al. [13] employed a tree structure in their taxonomy, which encompassed five primary categories: attack vector, operational impact, defense, informational impact, and

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

target (AVOIDIT). The authors placed significant emphasis on the classification of attacks, primarily focusing on the attack processes. Consequently, their attention did not extend to mitigating the attacks or incorporating other crucial aspects, such as the security properties at risk (confidentiality, integrity, and availability).

Chan et al. [14] exposed a taxonomy that provides a way to understand attacks on a new classification. Attacks were grouped and classified based on three parameters: the web services layer, attack methodology, and effect. The proposed taxonomy can provide the flexibility to classify new web service attacks in a SOA environment only.

In [27] the authors focus on providing a comprehensive understanding of security attacks and risk assessment in the context of cloud computing. The authors aim to develop a taxonomy that categorizes various security attacks in the cloud environment and propose a risk assessment framework for evaluating the associated risks.

Yassine et al. [28] aim to provide an organized and structured taxonomy that categorizes various types of threats faced by web applications. The authors develop a taxonomy that classifies web application threats into different categories, considering factors such as the attack vector, the targeted component, and the impact of the threat.

Prinetto and Roascio [29] present a thorough and structured taxonomy of hardware vulnerabilities and attacks. This research addresses hardware trust and the authenticity of components, proposing a comprehensive taxonomy of hardware vulnerabilities and attacks, classifying them based on their domain, nature, target, goal, implementation method, and domain.

Previous research describes different approaches and features of taxonomies created to analyze vulnerabilities in web services and applications. This research has several benefits compared to the research cited above:

1. A specific approach to the challenges of classifying attacks in web services.
2. The proposed methodology considers the constantly evolving nature of attacks, recognizing that attackers are always generating new techniques and attack schemes.
3. The inclusion of multiple categories and attacks provides a wide range of attacks categorized based on attack properties, making them easy to classify.
4. The application of attack trees allows us to model and visualize the sequence of attacks and the interrelationships between them, providing a graphical representation and understanding of how they can be mitigated and counteracted.
5. Properly classifying attacks using this methodology helps researchers determine which countermeasures are most effective for each type of attack. By providing this information, the research makes it easier to select and apply the appropriate countermeasures to protect web services or another technology.

III. Proposed Methodology

One of the difficulties in finding vulnerabilities in web services during the execution phase is identifying attack scenarios. These scenarios are time-consuming to find and set up a bank of relevant attacks and automate them according to the test environment.

The objective of this research is to identify new types of attacks against web services following the steps outlined in Fig. 2. The rest of the section describes the steps of the attack taxonomy methodology.

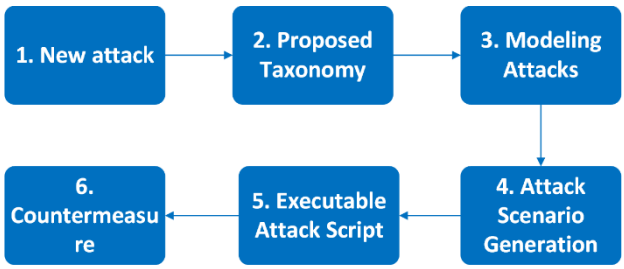

Fig. 2. Attack taxonomy methodology applied to web services.

A. New Attack

There are two ways to identify new vulnerabilities in web services: detecting existing attacks in the environment or identifying new vulnerabilities in the protocol stack used by the web service. In the first case, the researcher has to make use of honeypots and websites that publish attacks. In the second case, the researcher must use black-box or white- box techniques to identify unknown vulnerabilities.

The first step is to define an approach to systematically obtain new attacks, or variations thereof, that is successful enough to be classified in the attack taxonomy in the next step. If the attack is like another known attack, the researcher must decide whether to create a new item or a subclassification item, based on the characteristics described in Table I.

B. Attacks Taxonomy Against Web Services

Due to the magnitude of threats against web services today, it is possible to use different taxonomies to classify these attacks. This allows a better understanding of the potential threat and facilitates the application of possible countermeasures to each of these. For example, Landan in [15] categorizes security challenges as threats, attacks, and security problems divided into two levels:

- Service-level threats, also known as process-level attacks, are carried out in two web service protocol stacks (service discovery (UDDI) and service description (WSDL)) as well as SOAP message processing. Among the attacks against WSDL and UDDI are malicious code injection, phishing, denial-of-service (DoS), XML spoofing schema, session hijacking, and others.
- Message-level threats executed on the other two protocol stacks: transport protocol (HTTP, SMTP, FTP) and message protocol (XML, WS-Addressing, SOAP). These protocols enable attackers to execute various malicious activities, including fault injection attacks, message forwarding, message validation manipulation, interception, and compromising message confidentiality, among other potential exploits.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

The problem with the classification above consists of the inability to clearly define which protocol stack the attack belongs to. Another way to classify the attacks comes through the security properties affected. These properties are confidentiality (C), data integrity (I), availability (A) and access control issues (AC). In this case, the vulnerability occurs when an attack violates more than one property, e.g., a WSDL Scanning attacks violate confidentiality and access control properties because it looks for vulnerabilities in the WSDL.

The design of the following taxonomy allows researchers to explore many types of vulnerabilities and use specific features of each attack to analyze how to affect web services, as shown in Fig. 3, of a tree model of attacks against web services composed of 33 attacks classified into five categories.

The selection of the number of attacks and categories in this research is justified based on the security properties affected by the attacks, references to many kinds of known attacks, level of attack (WSDL or SOAP), impact level according to the OWASP, type of attack that can concentrate various types of known attacks, and possible countermeasures according to the WS-I. It is also possible to increase both the number of attacks and the categories of attacks.

These threats were collected from several studies that examined potential vulnerabilities, even with the use of security specifications such as WS-Security or WS-Trust. The Table I describes these attacks and its features, using the following criteria:

- Attack Type: Denial-of-Services (DoS), Brute Force (BrF), Spoofing (Spo), Flooding (Flo) and Injection (Inj).
- Attack: number and name.
- References.
- Security properties affected: confidentiality (C), integrity (I), availability (A), access control (AC).
- Attack level: service level (WSDL), and message level (SOAP).
- Sending requests to execute the attack, e.g. 1 or 1+ (at least one or more messages) or n (many messages).
- According to the OWASP [9], the impact level for successful attacks in a business environment is classified as low, medium, or high risk.
- Possible countermeasures, according to the Web Services Interoperability Organization (WS-I) [16].

Below, it is described each of these types of attacks against web services, as depicted in Fig. 3.

1) Denial-of-Service Attacks (DoS): It is an attempt to make the system resources unavailable to its users. This is not an invasion of the system, but an invalidation by overload. DoS attacks are typically carried out in two ways: i) forcing the victim system to reboot or consume all resources, such as memory or processing overhead, so that it cannot provideits service; and ii) interfering with communication between users and the target system in order to impair its functioning. It is composed of replay attacks, oversized payloads, coercive parsing, oversize cryptography, attack obfuscation, XML bombs, invalidated redirects and forwards, SOAP attachment, and schema poisoning.

2) Brute Force Attacks: The strategy used by this attack is to break the security system's encryption, consisting of exploring all possible key combinations in a cipher algorithm until the correct key is found. These attacks, since they use the method of trial and error, are very expensive in computational time. Web services that are vulnerable to this attack are insecure cryptographic storage, broken authentication, and session management.

3) Spoofing Attacks: This attack consists of a set of identity theft techniques in which the hacker successfully masquerades as another in order to falsify data and thereby gain an illegitimate advantage, i.e., web service resources. It is composed of SOAPAction, WSDL scanning, insufficient transport layer protection, WS-Addressing, middleware hijacking, metadata, security misconfiguration, unauthorized access, routing detours, attacks on WS-Security, attacks on WS-Trust, and malicious content.

4) Flooding Attacks: It is characterized by trying to cause a breakdown in the target system by providing more workload than the system can support. A flooding attack uses traffic redirection techniques and output port modification. It is composed of instantiation flooding, indirect flooding, and BPEL state deviation.

5) Injection Attacks: This type of attack involves intercepting and manipulating messages. The attackers aim to exploit vulnerabilities on the server-side to execute malicious commands, gain unauthorized access to data, and take control of the server. Injection attacks encompass various subtypes, such as XML injection, SQL injection, XPath injection, cross-site scripting (XSS), cross-site request forgery (XSRF), fuzzing scans, invalid types, parameter tampering, malformed XML, and Frankenstein messages (timestamp tampering).

In the first two steps, as shown in Fig. 2, the researcher can verify whether or not the new attack falls into any of the categories and types of attacks since there is a high possibility that it will be ruled out as a known attack and move on to the countermeasures phase

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

TABLE I. ATTACKS AGAINS WEB SERVICES AND THEIR COUNTERMEASURE.

Type	*Attack*	*References*	*Properties*	*Attack Level*	*Request*	*Impact*	*Counterme. (use)*
DoS	A.01. Replay Attack	[17], [18]	A	Message	n	Low	-
DoS	A.02. Oversize Payload	[5], [18], [19]	A	Message	1	Low	-
DoS	A.03. Coercive Parsing (Recursive Payloads)	[5], [18], [19]	A	Message	1	Medium	-
DoS	A.04. Oversize Cryptography	[5], [18], [19]	A	Message	1	Medium	-
DoS	A.05. Attack Obfuscation	[5], [18], [19]	A	Message	1	Low	-
DoS	A.06. XML Bomb	[17], [18]	A	Message	1	Low	-
DoS	A.07. Invalidated Redirects and Forwards	[5], [18]	C, A	Message	1+	Medium	-
DoS	A.08. Attacks through SOAP Attachment	[17], [18], [20]	I, A, AC	Message	1	High	-
BrF	A.09. Insecure Cryptographic Storage	[5], [21]	C, I, A	Service	1+	Medium	-
BrF	A.10. Broken Authentication and Session Mgmt	[5], [21]	AC	Service	1+	High	WS-Security
Spo	A.11. WSDL Scanning	[19]	C, AC	Message	1	Medium	WS-Security
Spo	A.12. Insufficient Transport Layer Protection	[5]	C	Message	1	High	WS-Security
Spo	A.13. Metadata Spoofing	[19]	All	Message	1+	Medium	-
Spo	A.14. WS-Addressing Spoofing	[19]	C, A	Message	2	Medium	WS-Security
Spo	A.15. Middleware Hijacking	[19], [21]	A, AC	Service	2+	Medium	-
Spo	A.16. SOAPAction	[19], [20]	AC	Message	1	Medium	WS-Security
Spo	A.17. Security Misconfiguration	[5]	All	All levels	1+	Medium	WS-Security
Spo	A.18. Routing Detours	[21], [22]	C, I	Message	1+	Medium	WS-Security
Spo	A.19. ATK on WS-Trust, WS-Secure Conversat.	[20]	I, AC	Message	2+	Medium	WS-Security
Spo	A.20. Attack on WS-Security	[20]	AC	Message	2+	High	WS-Security
Flo	A.21. Instantiation Flooding	[19]	A	Service	1	High	-
Flo	A.22. Indirect Flooding	[19]	A, AC	Service	2+	High	-
Flo	A.23. BPEL State Deviation	[19]	A	Service	1	High	-
Inj	A.24 XML Injection	[5], [19]	I	Message	2+	Low	WS-Security
Inj	A.25 SQL Injection	[5], [21]	I	Message	2+	High	WS-Security
Inj	A.26 XPath Injection	[17]	I	Message	2+	High	WS-Security
Inj	A.27 Cross-site Scripting (XSS)	[5], [21]	I	Message	2+	Medium	WS-Security
Inj	A.28 Cross-site Request Forgery (XSRF)	[5], [21]	AC	Message	2+	Medium	WS-Security
Inj	A.29 Fuzzing Scan	[17]	I	Message	n	Low	WS-Security
Inj	A.30 Invalid Types	[17], [19], [21]	I	Message	2+	Low	WS-Security
Inj	A.31 Parameter Tampering	[5], [17], [21]	I	Message	2+	Low	WS-Security
Inj	A.32 Malformed XML	[17]	I	Message	2+	Medium	WS-Security
Inj	A.33 Frankenstein Message (Modify Timestamp)	[20]	I, AC	Message	1+	Medium	WS-Security

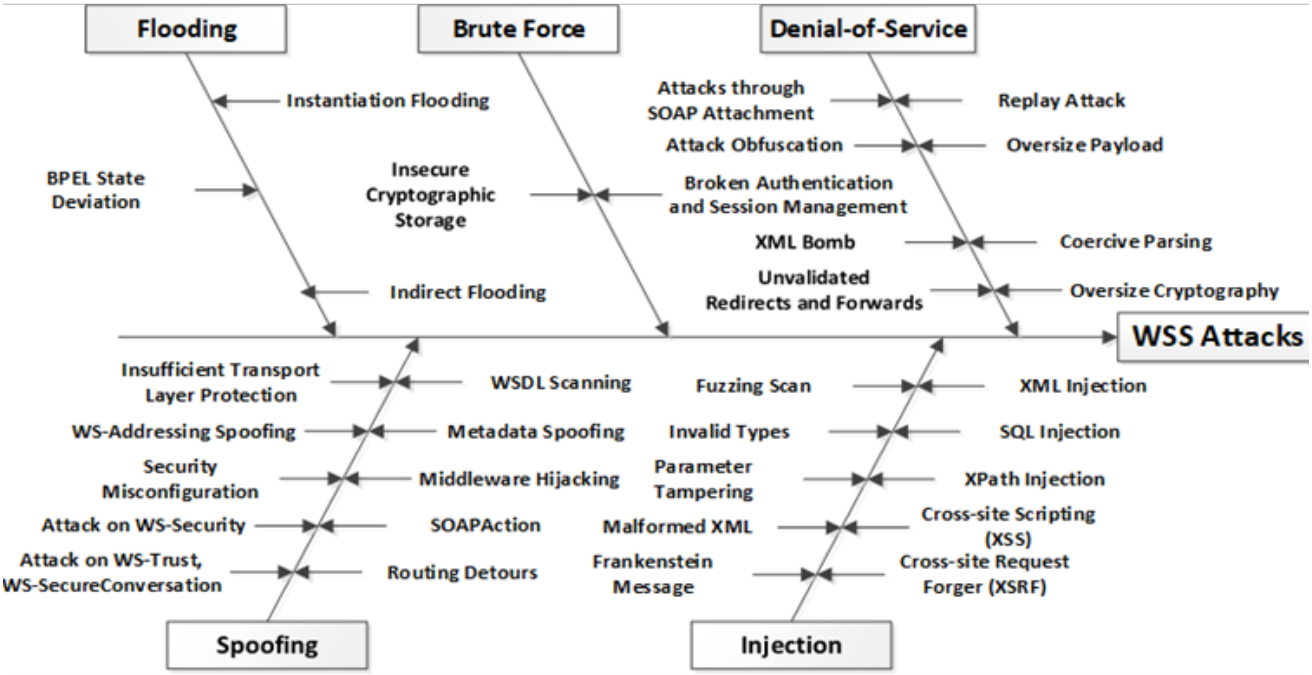

Fig. 3. Web services security attacks composed of 33 attacks classified into 5 categories

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

One of the challenges in reducing potential vulnerabilities in web services through detecting malicious or accidental flaws is determining the appropriate attack scenarios. At this point, it is possible to automate many types of attacks according to the test. This technique is described in the next step.

C. *Modeling Attacks*

To model the attacks, the researcher makes use of a modeling tool like SecureITree [23]. This tool allows building attack trees according to the attributes proposed in Subsection III-B.

Initially, the attacks are classified according to some features described in Table I. These features allow generating an attack tree, as seen in Fig. 3, to allow the researcher to select an attack type, attack its main security properties, and identify other characteristics that allow a successful attack.

A starting point is to develop a set of questions that allow the researcher to identify if they can carry out the attack on their systems. Next, it is suggested to ask the following questions:

- Does the researcher have the ability (knowledge) to carry out the attack?
- Does the researcher get to emulate the attack scenario through a tool or platform such as SoapUI [24]?
- Does the web service satisfy the required features to carry out the attack?
- Is WS-Security or another standard protecting the web service from this attack (impossible) or not (possible)?

The researcher must answer the questions for each attack. If the four questions are affirmative, i.e., possible <P, P, P, P>, the attack can be executed. There is a special case when it is necessary to use WS-Security to execute a certain type of attack, such as Oversize Cryptography or Attack Obfuscation. In this case, the third attribute will have the value of P (possible) only when the web service executes the security standard that allows the attack execution. Otherwise, it will be impossible (I). Besides, the security standard must not prevent the execution of the attack, or the fourth attribute will be impossible (I).

Once the four questions for the attack taxonomy (see Table I) have been evaluated with logical values (possible and impossible), the attack tree will be obtained and it can begin to look for vulnerabilities in web services. The output can be seen in Fig. 4.

After ensuring the requirements of the four attributes of the selected attacks, the researcher can generate the scenarios for each of them

```
OR  1 – Goal: Attack web services
    OR  1.1    Breach Confidentiality
        OR  1.1.2   A.12. Insufficient Transport Layer Protection <P, P, P, P>
    OR  1.2    Breach Integrity
        OR  1.2.2   A.24. XML Injection <P, P, P, P>
            1.2.4   A.26. XPath Injection <P, P, P , P>
            1.2.5   A.27. Cross-site Scripting (XSS) <P, P, P, P>
            1.2.6   A.29. Fuzzing Scan <P, P, P, P>
            1.2.7   A.30. Invalid Types <P, P, P, P>
            1.2.9   A.32. Malformed XML <P, P, P, P>
            1.2.10  A.33. Frankenstein Message: Modify Timestamp <P, P, P, P>
    OR  1.3    Breach Availability
        OR  1.3.1   A.01. Replay Attack <P, P, P, I>
            1.3.2   A.02. Oversize Payload <P, P, P, I>
            1.3.3   A.03. Coercive Parsing <P, P, P, I>
            1.3.4   A.04. Oversize Cryptography <P, P, I, I>
            1.3.5   A.05. Attack Obfuscation <P, P, I, I>
            1.3.6   A.06. XML Bomb <P, P, P, I>
    OR  1.4    Breach Access Control (Authentication and Authorization)
        OR  1.4.4   A.08. Attacks through SOAP Attachment <P, P, I, I>
```

Fig. 4. Web services attack tree.

D. *Attack Scenario Generation*

The attack scenarios are produced automatically through the attributes used in the attack tree. The scenarios obtained in this section can be used to create a library of attacks useful for testing other web services, protocols, or systems, facilitating their reuse. Fig. 5 describes an example of an attack scenario for the XML Injection attack, using information obtained in [5], and [19] about the operation of the attack and the requirements.

The result of this step is the generation of attack scenarios described in textual language, which is on the same level of abstraction as the attack tree. This descriptive format proves valuable for test analysts and security experts due to its ease of configuration. However, it is important to note that this type of description is not directly processable by a tool or automated system.

Analysts must perform a set of refinement steps in order to transform attack scenarios in textual language into a script executable by the attacker's preferred tool, such as SoapUI [24].

```
1   Objective:  XML Injection attack seeks to modify XML structure.
2   Preconditions:
3               - The client sends a request to Web Services via a SOAP message.
4               - The client does not use a secure communication scheme.
5               - The WSDL describes at least one parameter.
6   Attack:
7           AND 1. If required a WSDL to send a message.
8               2. If it contains the searched <String> string or tag.
9               3. Modify the request parameters by a non-default message.
10              4. Generate the new SOAP message.
```

Fig. 5. XML Injection features for the Attack Scenario.

E. *Executable Attack Script*

The generation of executable attack scripts is important to experimentally validate the vulnerability found. The messages exchanged between the server and the client must be monitored and collected to determine if the attack was effective (true positive) or not (true negative), as well as if it is necessary to modify the attack script.

It is recommended the use of soapUI [24] or another security testing tool to automate the attacks and allow to monitor in real time, as seen in the Fig. 6.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

```
1:  HTTP/1.1 200 OK
2:  Cache-Control: private, max-age=0
3:  Content-Length: 455
4:  Content-Type: text/xml; charset=utf-8
5:  Server: Microsoft-IIS/7.0
6:  X-AspNet-Version: 4.0.30319
7:  X-Powered-By: ASP.NET
8:  Date: Mon, 26 Oct 2021 10:12:06 GMT
9:
10: <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
11: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
12: xmlns:xsd="http://www.w3.org/2001/XMLSchema">
13:   <soap:Body>
14:     <GetISOCountryCodeByCountyNameResponse xmlns="http://www...X.NET">
15:       <GetISOCountryCodeByCountyNameResult>&lt;NewDataSet /></GetISOCountryCodeByCountyNameResult>
16:       </GetISOCountryCodeByCountyNameResult>
17:     </GetISOCountryCodeByCountyNameResponse>
18:   </soap:Body>
19: </soap:Envelope>
```

Fig. 6. The vulnerability was discovered through an injection attack and the execution of the string lt;NewDataSet.

Finally, it is important to reduce potential false positives and false negatives through the development of rules that allow for determining when there is a confirmed attack.

F. Countermeasures

During the 2000s and 2010s, one of the most widely used options against cross-site scripting (XSS) or XML Injection attacks was the use of security protocols developed by OWASP, such as WS-Security or XML Encryption [25]. In recent years, the use of Web application firewalls (WAF) [26] became popular. This tool is a specialized application firewall designed to filter, monitor, and block HTTP traffic to and from a web service, with a particular emphasis on Layer 7 applications. It serves as an effective defense against attacks that exploit well-known vulnerabilities in web applications, including SQL injection, cross-site scripting (XSS), XPath Injection, and malformed XML. By leveraging its capabilities, this tool acts as a proactive shield, preventing the successful exploitation of known vulnerabilities of a web application.

Although WAF can block different types of attacks, it still requires the help of the authentication and encryption protocols offered by SOAP Foundation.

This section describes some security mechanisms to protect web services (see Fig. 7) against many kinds of web service attacks like WS-Security or XML Encryption taking into account three aspects: (i) message authentication in order to make sure that a transaction between the server and its client is legitimate; (ii) confidentiality to protect exchanged messages against interception by an unauthorized third party; and (iii) integrity of messages sent between server and client in order to remain unaltered.

X = Partial protection of this kind of attacks. O = Reduce the impact of this kind of attacks.	DoS	Brute Force	Flooding	Spoofing	Injection	XML Injection
XML Encryption			O	X	O	
XML Signature			X	X	O	X
Security Tokens			X	X	O	O
WS-Addressing					O	O
SSL/TLS [end to end]			X	X	O	X
HTTP Authentication			X	X	O	

Fig. 7. Security mechanisms to protect web services.

Some security mechanisms to protect web services are described below.

1) WS-Security (WSS): This standard contains specifications that guarantee the confidentiality and integrity of messages and user authentication. It inserts a layer over the SOAP message to build more secure and robust services with broad interoperability. In addition to being a solid and open security model, WS-Security is fast-developing, allowing users to encrypt XML documents and secure sessions between two or more parties [25] using other specifications such as Security Tokens, XML-Encryption, and XML-Signature.

2) Security Tokens: The Security Token is a security specification for providing authentication and authorization in web services, providing access rights to application servers. It makes use of the tag to provide different credential types, such as identification by user/password, to more complex ones, based on certificates such as X.509 and Kerberos [25].

3) XML Encryption (XML-Enc): This specification provides confidentiality and authentication to the web service by encrypting information between the parties. It makes use of the <EncrytedData> tag to use the encryption key. Thus, users who do not have the key will not have access to the message and its contents. The technology [25] allows the use of multiple cryptographic keys for different parts of a message. In turn, the same message can have several receivers, and each receiver only has access to its own parts of the message.

4) XML Signature (XMLDSig or XML-Sig): This pattern makes use of the <Signature> tag and is used for two purposes: it validates Security Token credentials and verifies that messages are not modified during transmission to ensure their integrity. Verification of credentials is done using the signature in combination with the certificate to ensure that the user is who they claim to be [25]. Similar to XML Encryption, XML-Sig allows users to sign certain portions of the message.

IV. Attack Taxonomy Methodology Applied to Session Fixation

A recently published attack [30], called Session Fixation, is used to apply the attack taxonomy methodology.

A session fixation attack occurs when an attacker manipulates a user's session ID to a specific value, granting them unauthorized access. Attackers employ various techniques, such as cross-site scripting exploits or reusing HTTP requests, to achieve session fixation. The attack typically unfolds in two steps: the attacker fixes the victim's user session ID and then the victim unknowingly logs in, unknowingly exposing their online identity. With the fixed session ID value, the attacker can subsequently hijack the victim's user identity and gain unauthorized control.

In this way, the attacker modifies the message (integrity) and attempts to impersonate the identity of the victim (access control) by sending one or more messages at the SOAP message level. According to OWASP [9], the impact level of a successful attack is classified as medium to high risk for reaching a level of authentication in the system. Furthermore, it can be classified as a spoofing attack in the subsection III-B3 due to its similarity to A.17 Security Misconfiguration, described in Table I.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

LATIN-AMERICAN JOURNAL OF COMPUTING (LAJC), Vol XI, Issue 1, January 2024

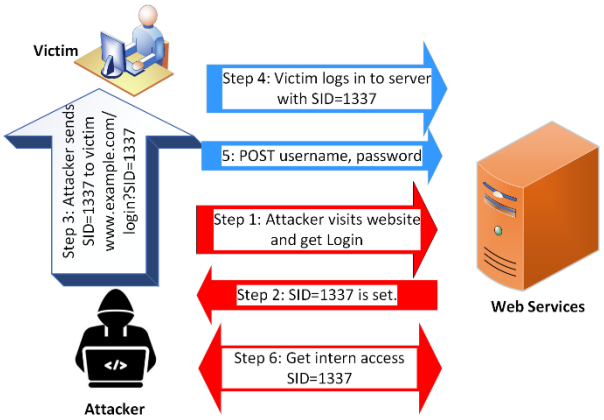

Fig. 8. Exemplified Session Attack in web services

The Session Fixation attack would fall under the category of "Spoofing" (Spo). The main goal of a Session Fixation attack is to trick the system into accepting a fake session or identity as valid. This involves intentionally manipulating or fixing a user's session ID and then impersonating that identity. This type of attack is considered a form of identity theft and falls under the category of "Spoofing".

In the modeling phase, the attacker gets <p, i, p, p> because:

- The attacker has knowledge to implement the attack (possible), as shown in Fig. 8.
- SoapUI does not emulate the Session Fixation attack (impossible). In this case we use another tool.
- I have access to web services that require authentication and do not use a web application firewall or other protection standard (possible).
- WS-Security, XML-Encryption, XML-Signature, and Security Tokens can protect against this attack (possible).

Since the attacker cannot ensure the requirements for the four attributes described above, the Session Attack must be manually programmed to generate the attack scenario. In generating the attack scenario, the attacker can use Burp Suite[3], OWASP ZAP[4], WebScarab[5], or BeEF[6] to simulate the type of attack. Next, we make the textual description of the Session Fixation attack described in Fig. 9.

In the next step, we generate the script for the Session Fixation attack using cookies in a web server authentication scenario:

1. Attacker: Generates a fixed session ID: "fixed_session_id".
2. Victim: The victim visits the targeted website without having previously logged in.

```
1   Objective:  Take control of a user's session on a website.
2   Preconditions:
3       - WS must use sessions to authenticate and maintain user state.
4       - WS must use a session ID mapping mechanism, such as cookies, URLs, or
          hidden fields in forms.
5       - The attacker needs to know the mechanism used to assign and manage
          session IDs in the WS.
6       - The attacker must have the ability to manipulate or force the user to use a
          specific session ID before they log in to the WS.
7   Attack:
8   AND 1. The attacker identifies the mechanism used by the WS to assign and
          manage session IDs.
9       2. The attacker generates a specific session ID and sets it, either by
          manipulating the allocation mechanism or tricking the user into using a
          predefined session ID.
10      3. The attacker sends the pinned session ID to the legitimate user or
          otherwise makes it accessible for the user to use when logging into the WS.
11      4. The user logs into the WS using the session ID provided by the attacker.
12      5. Once the user is logged in, the attacker uses the same set session ID to
          access the user's active session and carries out malicious actions on its
          behalf.
```

Fig. 9. Session Fixation features for the Attack Scenario.

3. The victim receives the email and clicks on the malicious link. The victim's browser opens the target website and sets a session cookie with the fixed session ID provided by the attacker.
4. The attacker now knows the session ID that the victim will use when logging in.
5. The victim continues to use the website and decides to log into her account. The victim's browser sends an HTTP POST request to the server with the login credentials and the session cookie containing the fixed session ID.
6. Server (response at HTTP protocol level):
   - The server receives the POST request with the login credentials and the session cookie.
   - The server validates the credentials and verifies the session cookie.
   - Because the session cookie is valid and contains the fixed session ID, the server considers the session to be legitimate and authenticates the user as the victim.

In this way, the attacker managed to set the victim's session ID before the victim logged in. As a result, the attacker can impersonate the victim and access the victim's account without having to provide the correct login credentials.

At the HTTP protocol level, the POST request sent by the victim's browser to the server would include the login form data and headers, such as the following:

```
1   POST /login HTTP/1.1
2   Host: www.bank.com
3   Cookie: session_id=fixed_session_id
4   Content-Type: application/x-www-form-urlencoded
5   Content-Length: 23
6
7   username=victim&password=secret
```

Fig. 10. The session fixation attack was successful through an HTTP protocol request. The attacker has access to the user name and password of the victim.

[3] https://portswigger.net/burp
[4] https://owasp.org/www-project-zap/
[5] https://github.com/OWASP/OWASP-WebScarab
[6] https://beefproject.com/

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

The server would process the request and authenticate the user based on the session cookie containing the fixed session ID provided by the attacker.

There are several countermeasures that you can use to mitigate a Session Fixation attack. Since this attack undermines the properties of integrity (I) and access control (AC), Fig. 7 of security mechanisms suggests the use of a combination of techniques to reduce the impact of said attack, made up of XML Encryption, XML Signature, Security Tokens, use of SSL, and HTTPS. We can also establish a session expiration policy, implement anti-CSRF tokens, and monitor and log suspicious activities, both on the server and on the compromised computer.

V. Conclusions

The attack taxonomy methodology contributed to the development of security research in web services by describing the security properties affected, the level at which they develop, and other features.

This methodology can be used to explore many types of vulnerabilities and use specific features of each attack, like Session Attack. The objective is to analyze how an attack can affect web services, in addition to creating new attacks and selecting possible countermeasures.

In this way, this research described five categories of web services attacks (brute force, spoofing, flooding, denial-of-services, and injection attack types) along with thirty-three (33) attacks to provide a state of the art.

As shown in Table I, this taxonomy allows researchers to classify new attacks based on properties (integrity, availability, confidentiality, and access control), level of attack (WSDL or SOAP), amount of exchange of messages, or level of impact according to the OWASP Top Ten.

Furthermore, a correct classification or grouping of an attack will allow researchers to more easily determine which potential countermeasures to employ.

In the future, it is proposed to apply this systematic methodology to different technologies. Furthermore, it is possible to combine this methodology with malware attacks like botnets.

Acknowledgment

The author would like to thank Eliana Martins (LSD Laboratory), and Paulo Lício de Geus (LASCA Laboratory) from the Computing Institute of University of Campinas (UNICAMP).

References

[1] R. Derbyshire, B. Green, D. Prince, A. Mauthe, and D. Hutchison, “An analysis of cyber security attack taxonomies,” in 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 153– 161, IEEE, 2018.

[2] C. Ferris and J. Farrell, “What are web services?,” Communications of the ACM, vol. 46, no. 6, p. 31, 2003.

[3] “Web services tutorial.” https://www.tutorialspoint.com/webservices/what_are_web_services.htm, 2022.

[4] I. siddavatam and J. Gadge, “Comprehensive test mechanism to detect attack on web services,” in 2008 16th IEEE International Conference on Networks, pp. 1–6, 2008.

[5] D. Wichers and J. Williams, “Owasp top-10 2021.” https://owasp.org/Top10/.

[6] C. Top and M. Dangerous, “2021 cwe top 25 most dangerous software weaknesses, 2021.” https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html.

[7] H. Yuan, L. Zheng, L. Dong, X. Peng, Y. Zhuang, and G. Deng, “Research and implementation of web application firewall based on feature matching,” in International Conference on Application of Intelligent Systems in Multi-modal Information Analytics, pp. 1223–1231, Springer, 2019.

[8] B. Jagruti, P. Nidhi, and D. Pandya, “A survey on webservice security techniques,” in 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1–5, 2018.

[9] O. B. Fredj, O. Cheikhrouhou, M. Krichen, H. Hamam, and A. Derhab, “An owasp top ten driven survey on web application protection methods,” in Risks and Security of Internet and Systems (J. Garcia-Alfaro, J. Leneutre, N. Cuppens, and R. Yaich, eds.), (Cham), pp. 235–252, Springer International Publishing, 2021.

[10] W. Ahmad, Z. Hayat, B. Zafar, F. A. Khan, F. Din, and I. Shah, “A survey on taxonomies of attacks and vulnerabilities in computer systems,” International Journal of Computer Science and Telecommunications, vol. 3, no. 5, pp. 93–97, 2012.

[11] S. L. Hansman, “A taxonomy of network and computer attack methodologies,” 2003.

[12] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, “Security issues in iiot: A comprehensive survey of attacks on iiot and its countermeasures,” in 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), pp. 124–130, 2018.

[13] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta, and Q. Wu, “Avoidit: A cyber attack taxonomy,” in 9th Annual Symposium on Information Assurance (ASIA’14), pp. 2–12, 2014.

[14] K. F. P. Chan, M. Olivier, and R. P. van Heerden, “A taxonomy of web service attacks,” in Proceedings of the 8th International Conference on Information Warfare and Security: ICIW, p. 34, 2013.

[15] M. I. Ladan, “Web services: Security challenges,” in 2011 World Congress on Internet Security (WorldCIS-2011), pp. 160–163, IEEE, 2011.

[16] R. K. K. Meduri, “Webservice security,” in Webservices, pp. 119–172, Springer, 2019.

[17] P. Nandan, Mastering SoapUI. Packt Publishing Ltd, 2016.

[18] G. A. Jaafar, S. M. Abdullah, and S. Ismail, “Review of recent detection methods for http ddos attack,” Journal of Computer Networks and Communications, vol. 2019, 2019.

[19] M. Jensen, N. Gruschka, and R. Herkenhöner, “A survey of attacks on web services,” Computer Science-Research and Development, vol. 24, no. 4, pp. 185–197, 2009.

[20] V. Patel, R. Mohandas, and A. R. Pais, “Attacks on web services and mitigation schemes,” in 2010 International Conference on Security and Cryptography (SECRYPT), pp. 1–6, IEEE, 2010.

[21] A. Singh, A. Sharma, N. Sharma, I. Kaushik, and B. Bhushan, “Taxonomy of attacks on web based applications,” in 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), vol. 1, pp. 1231–1235, 2019.

[22] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, “Assessing lightweight virtualization for security-as-a-service at the network edge,” IEICE Transactions on Communications, vol. 102, no. 5, pp. 970–977, 2019.

[23] A. T. Limited, “Attack tree modeling.” https://www.amenaza.com/, 2022.

[24] S. Mehta, G. Raj, and D. Singh, “Penetration testing as a test phase in web service testing a black box pen testing approach,” in Smart Computing and Informatics, pp. 623–635, Springer, 2018.

[25] S. Abidi, M. Essafi, C. G. Guegan, M. Fakhri, H. Witti, and H. H. B. Ghezala, “A web service security governance approach based on dedicated micro-services,” Procedia Computer Science, vol. 159, pp. 372– 386, 2019.

[26] “Web application firewall.” https://owasp.org/www-community/Web_ Application_Firewall, 2018.

[27] S. Akshaya, M., and G. Padmavathi. "Taxonomy of security attacks and risk assessment of cloud computing." Advances in Big Data and Cloud Computing: Proceedings of ICBDCC18. Springer Singapore, 2019.

[28] S. Yassine, and Y. Maleh. "A systematic review and taxonomy of web applications threats." Information Security Journal: A Global Perspective 31.1 (2022): 1-27.

[29] P. Prinetto, and G. Roascio. "Hardware Security, Vulnerabilities, and Attacks: A Comprehensive Taxonomy." ITASEC. 2020.

[30] "8 critical web application vulnerabilities and how to prevent them." https://brightsec.com/blog/web-application-vulnerabilities/, 2022.

Marcelo Invert Palma Salas holds his Master in Computer Science (2012) and PhD candidate in Computer Science from the University of Campinas (Unicamp), Brazil. Systems Engineer (2005) from the Military School of Engineering. He belongs to the Security and Cryptography Laboratory of the UNICAMP Computer Institute. Professor of the Computer Science Career at the Universidad Mayor de San Andrés, La Paz, Bolivia. Researcher in artificial intelligence (machine learning and deep learning), malware analysis, detection, and classification, and network security over Tor Network. Cybersecurity specialist, full-stack developer, Linux and Python server administrator. Awards winner of IEEE Richard E. Merwin Award (2013), IEEE Theodore W. Hissey Award (2009), Outstanding Researcher Award from the Bolivian Government (2016). Winner of the FRIDA Grant for the project "Tor Protection against malicious traffic" (2016). IETF 99 Grant in Prague (2017), Cheva Rep. to participate in DOTS communications, Internet Society, etc. IEEE member, TBB (Your Scholarship Bolivia). Reviewer of the journals IEEE Latin America and CLEI, among others.

ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2024
DOI: 10.5281/zenodo.10402238

AUTHORS

Marcelo Invert Palma Salas

Marcelo Invert Palma Salas holds his Master in Computer Science (2012) and PhD candidate in Computer Science from the University of Campinas (Unicamp), Brazil. Systems Engineer (2005) from the Military School of Engineering. He belongs to the Security and Cryptography Laboratory of the UNICAMP Computer Institute. Professor of the Computer Science Career at the Universidad Mayor de San Andrés, La Paz, Bolivia. Researcher in artificial intelligence (machine learning and deep learning), malware analysis, detection, and classification, and network security over Tor Network. Cybersecurity specialist, full-stack developer, Linux and Python server administrator. Awards winner of IEEE Richard E. Merwin Award (2013), IEEE Theodore W. Hissey Award (2009), Outstanding Researcher Award from the Bolivian Government (2016). Winner of the FRIDA Grant for the project "Tor Protection against malicious traffic" (2016). IETF 99 Grant in Prague (2017), Cheva Rep. to participate in DOTS communications, Internet Society, etc. IEEE member, TBB (Your Scholarship Bolivia). Reviewer of the journals IEEE Latin America and CLEI, among others.