Proceedings of the Digital Privacy and Security Conference 2022 | 10.11228/dpsc.04.01

Benefits, Issues and Best Practices of using Web Services

Rui Rebelo

Lusófona University of Porto, Portugal
ruypedro2011@hotmail.com

Abstract. Over the years, mobile devices have become an acquirement on our daily lives. Compared to other devices that can have access to web services, mobiles had a big growth in the last few years. With this being said, we can conclude that hackers can easily reach or steal an enormous amount of information and use it to manipulate or threat other people. Web services are a solution used to integrate systems and it can also be useful when it comes to communication between different applications. This technology allows new applications to communicate with older ones and that systems developed in different platforms can be compatible. These systems are components that allow applications to send and receive data. Like every software or application web services also have failures.
In this paper I will study the main threats of IT security, involving mobile devices, as well as the Benefits, Issues and Best Practices of using Web Services, by choosing one practical case to show the using of web services, approaching some known flaws and measures taken as well as some fears and preventions.

Keywords: Benefits, Issues, Best Practices, Web Services, Threats, IT Security, Mobile Services.

1 Introduction

In the last couple of years, the use of the internet has had a big growth in our daily lives in all possible areas. The mobile devices are the most used to access the internet since it's easily portable and facility of acquisition. According to [1] a study that was made in a school in Pune City, 97% of the students use mobile devices, called smartphones, and only 3% do not use them.

As we all know, almost every daily needs can be done on mobile devices, such as talking to a distant person, accessing our emails, a document that was sent to us or even our bank account, among many other utilities that mobiles provide. Having that in mind, there are a lot of existing threats, not only of mobiles being the most used electronic devices but also because many people save personal information, such as credit card details, social security number, among other relevant information that can be used to harm others.

In section two we will mention some of the most used mobile operating systems
In section three, we will see the most important threats in the security area and whether these threats had a big growth during the last couple of years, based on previous studies.
In section four we will focus on the advantages and disadvantages, best practices of web services
And finally, a practical case were we can see how to make an XML Injection very easily.

2 Most used mobile operating systems

As we all know, in the beginning, when phone were news, they worked differently from what we are used to currently. They weren't that developed since they were used for phone calls only. As years gone by, these devices have evolved allowing for example gaming, taking pictures, listening to music, among any other activities that we thought not possible. But for this to be possible, all the mobile devices have the need to have installed an operating system. But what is an operating system?

Operating systems began to be used in computers to mediate between the hardware and the software, which means managing and controlling the resources and computing capability of a computer and allow users an interface to work with the physical computer's structure.[2] But in the beginning that interface was not "so good to the eyes", since it was just a simple command line where we executed lines of code to do the work. After a couple of years the first operating system appeared with a graphical interface and multitasking support, which is almost the same thing that we all have in our machines right now. And this leads us to another question, what mobile operating systems have in common with computer operating systems? In fact, those to different operating systems have a lot in common, because they were created to do the same task which is to mediate between the hardware and software. The only big difference is the size of the machine where they are implemented. As we also know, there is a big panoply of mobile operating systems but according to [1] when studies were made in a certain school 28,50% of the students said that they use Samsung which uses android as operating system, 21,62% said that they uses Apple's iPhone which uses IOS as operating system, which makes these two, the most used mobile operating systems. In the following we are going to talk a little bit about these two operating system's history.

According to [3], the android was initially created by Android Inc. and after that bought by Google. It was released for the first time in 2007 as Android Open Source Project. Android is a fairly young platform, its use takes place very quickly. Each major release is named in alphabetic order after a dessert or sugary treat.

According to [4], it all started in 1976 in a small garage in Los Altos, California, with three names well known, being Steve Jobs, Steve Wozniak and Ronald Wayne. Initially, the name would not be the one as we know it for but iPhone OS instead. Later in June 2010, after Apple had made some big changes in their devices, for example, to implement a store called as App Store, allowing users to buy and download applications to their devices, which led to a name changing being it now IOS. But so user could download and install the applications some adjustments were needed so that the device was capable of handling a large amount of programs.

3 Mobile principal threats of IT Security

The easy access of these devices brought us the possibility to use them in isolated environments but also in network environments. It also brought us a lot of advantages as well as some disadvantages. For devices' network or system to be safe, it has to ensure confidentiality, integrity and availability which means, only some users of the system can access sensitive information. It assures us that the information was not modified or destroyed and also that the information will not be available to users who are not logged in. But all computed devices have threats in the security area and mobile devices are not an exception. But what is a threat in the computer area? A threat as the name suggests is something that can damage the device or steal some important information from your device, sometimes forcing people to pay a large amount of money to recover that information. According to [5] this activity is caused by a person to carry out criminal acts that would harm who they steal the information from.

With this type of activity users have to be cautious since they can download a virus or any other type of threat without knowing. These devices that we carry all the time in our pockets store a lot of important information. Even if the companies provide a lot of security, as mentioned before, users are still being exposed to a big number of possible attacks. According to [6] some of these attacks will be presented in the following:

(i) Phishing-in-the-app: We discovered one way that criminals can bypass the Play Market's source code checks was by not including anything malicious in the app itself. This app is nothing more than an embedded website, which make users believe they are having the perfect experience, not knowing that they are being a victim of phishing.

(ii) Supply chain compromise: Following a lead from an online message board, we discovered a Trojanized version of a legitimate app that had been included in the factory firmware. The original application, called Sound Recorder, was modified to include lines of code that were not necessary for its purpose. This additional code was used to intercept and secretly send SMS.

(iii) Cryptominer code in games or utilities: The SophosLabs team have, in the normal course of looking for mobile malware, encountered a significant jump in the number of apps that, without notifying the user, included cryptominer code in the app. The cryptominer is a way to earn digital coins that can be converted to real money. To earn the coins users need to solve mathematical puzzles. To do this quickly, hackers implement code in applications that users will download. The code would run whether or not the app itself was running, and functioned as a constant drain on the phone's battery.

(iv) Advertising click-fraud embedded in apps: Advertisement fraud is, surprisingly, one of the most profitable criminal enterprises nowadays, and mobile apps appear to be a key part of this subtle crime. This type of crime occurs frequently in mobile applications. Sometimes, when we are playing a game on our mobile device, advertising appears almost instantly without our meaning to. This type of advertising may contain some type of fraud that is not visible. This crime can be very profitable in monetary terms.

According to OWASP [7], companies should start the process of ensuring that their applications minimize ten risks. Below, we will present the ones we think are most important.

(i) Broken Access Control: Access control is used to make sure that users cannot act beyond their permissions. When this access control is broken the hacker will gain access to unauthorized information and will be able to change or destroy it.[8]

(ii) Cryptographic Failures: The first thing to determine is the level of protection the data should have. If this data doesn't have any kind of protection, it is much easier for hackers to have access to them. For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection, mainly if that data falls under privacy laws, for example, General Data Protection Regulation (GDPR).[9]

(iii) Injection: There are many types of injection like SQL, NoSQL, OS command, Object Relational Mapping (ORM), LDAP, and Expression Language (EL) or Object Graph Navigation Library (OGNL) injection. To detect if the application is vulnerable it's necessary to review the source code. Normally the injection is performed in the URL or in some input space that will be used for some query.[10]

(iv) Security Misconfiguration: An application is vulnerable if, for example, unnecessary features are enabled or installed, like unnecessary ports, accounts, or privileges that should be disabled.[11]

Proceedings of the Digital Privacy and Security Conference 2022 | 10.11228/dpsc.04.01

(v) Software and Data Integrity Failures: These failures are related to code and infrastructure that does not protect against integrity violations. This normally happen when the application uses plugins or libraries from untrusted sources. [12]

(vi) Server-Side Request Forgery: These failures occur whenever a web application searches for a remote resource without validating the user-supplied URL. It allows the hacker to force the application to send a crafted request to an unexpected destination.[13]

Other threats that can happen more frequently than we think is the loss of the device, its theft or even the accidental or malicious misuse of it. The accidental misuse of the device, can happen, for example, when we are exploring the device definitions and we change some important information, that should not be changed. Malicious use of the device can happen, for example, if we lend the device to someone and that same person changes something having access to the device in a way the owner does not know.

4 Benefits, issues and best practices of web services

Web services were implemented to solve the problem that was communication between applications that were developed a long time ago with new ones and also to communicate with others that were developed in different platforms. These services are used to allow changing data.

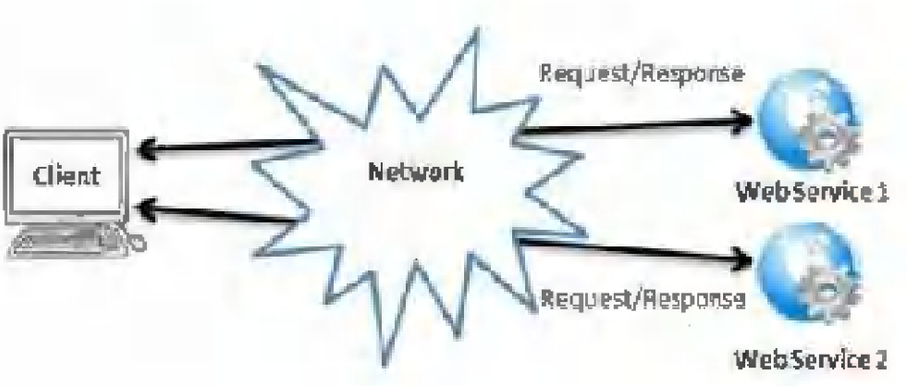

Fig. 1. Conceptual model of web services [14]

For the fact that applications have their own language, these services translate the communication to an intermediate format, being one of them XML. According to [15] XML is a simple text-based format for representing structured information: documents, data, configuration, books, transactions, invoices, and

a lot more. It was derived from an older standard format called SGML (ISO 8879), in order to be more suitable for Web use. As every system or application, web services also need to have many security measures like authentication, confidentiality, integrity and availability to ensure the best security and reliability.

4.1 Benefits

According to [16], web services provide several technological and business benefits, such as:

(i) Application and data integration
(ii) Versatility
(ii) Code re-use
(iv) Cost savings

As mentioned before, web services provide application and data integration because they allow applications to communicate between them and none of the parts need to know how the other is implemented or in what format the data is stored.

The versatility allows people to access the web service via a web-based client interface. The client can even combine data from multiple web services. For example a user being able to combine a product's price from different sellers and see which one is the least expensive and save some money.

The re-use of code can be very helpful because a lot of clients can use the same portion of code to do different tasks. Instead of having to create different services for different things, portions of a service can be simply re-used allowing the client to have some flexibility.

All three benefits mentioned before lead us to a fourth benefit that is cost savings. The need to create applications to connect with others can be very expensive and web services solve that issue easily so that cost is removed and money can be used to add additional value to the main application. Web services also take advantage of ubiquitous protocols and Web infrastructure that already exist in every organization, so they require little if any additional technology investment.

4.2 Issues

According to [17], the threats in web services can be through message or service level. The service level threats can involve UDDI, which is a protocol that defines how clients communicate with UDDI registers and it's also a specific set of replicated global records.[18] These threats can also involve WSDL which is a

Proceedings of the Digital Privacy and Security Conference 2022 | 10.11228/dpsc.04.01

language based on XML used to create services contracts. These contracts have all the information needed to create a client able to communicate with the web service.[19] Can also involve XML. In the following we are going to present some of the service level threats in a short form (the full text can be found at [17]):

(i) WSDL and UDDI attack: An attacker can access any public information available WSDL file and tamper with it. The former scans the WSDL file and exposes the operational information, ports, etc. The last one tampers the data and can even gain access to confidential information.

(ii) Malicious Code Injection and Identity Spoofing: This occurs when an attacker is able to inject malicious code and spoil the functionality of the service. Identity Spoofing occurs when the attacker takes off the identity of the service requester or the service provider.

(iii) XML Schema Tampering: The attacker can modify the original XML Schema and make it erroneous, causing the service to end up with failures.

(iv) Session Hijacking: An attacker can steal the token from the user and gain unauthorized access to the resources provided. This leads us to false request or replies and, because of that, we can say that the session was hijacked.

(v) Message Injection or Alteration: Messages traded between the server and the client can be modified or malicious messages can be added. This can provide the hacker many privileges.

(vi) Replay of Messages: The attacker captures a valid message traded between the user and the server and replays it later thus leading him to access sensitive information via unauthorized access. Usually this is the first step to hijack the session and tamper with the services.

(vii) Message Confidentiality and Eavesdropping: Interception of messages is always a threat to web services. Traditional security mechanisms are not sufficient to secure the web services against such threats.

4.3 Best Practices

As seen before, web services have benefits, as well as some issues. In the following we are going to present some general measures to use in web services and then some measures to solve the issues that can happen. As mentioned earlier, some of the best security practices are ensuring the confidentiality, integrity, and availability. These web services also need other type of security that is not just digital security. They also need a physical one, which means that these same web services need to be in data centers, which location is only known to a small number of people. Access to these data centers must be registered and accredited to be allowed.

According to[20], protecting credentials and session cookies is one of the most difficult tasks for a developer. In the following, some preventive measures that can be applied are going to be presented in a short form (the full text can be found on [20]):

(i) Using Secure Socket Layer: All the credentials should be stored in an encrypted form, for example, an attack to the database or some file system should not compromise credentials.

(ii) Expire Session after Inactivity: Forcing an automatic logout, after a reasonable time can be a good idea. This way an abandoned session will not be active for a long period of time and thus reduces the chance that a hacker has to find an active session.

(iii) Do Not Make Session Identifiers Viewable: This problem occurs in the GET method. The GET variables are always in the path string of the browser. With this, printing one of these pages will show always the identifier because most of the time, and most of the browser print the URL in the header. To prevent that we should use POST method.

(iv) Provide Secure Logout: As the name suggests we should provide the user a safe logout, that when used, the session will be inactivated. For example, when an users logs out, his session should be saved in the database and then deleted, or that same session should be marked as disabled. Using this when someone uses that session to login, the server can assume that this session is not valid.

(v) Use Strong Encryption on All Transmissions: The non use of encryption will turn the system almost completely insecure. The malefactor will be able to observe the communications done. But if the data is already encrypted this is not and issue since the data is rendered unreadable.

(vi) One-Time Cookie: One time cookies are generated by the reverse proxy server for each request of the user. This is a better alternative to authentication cookies that does not require volatile state in web browser.[21]

(vii) Schema Validation: The Schema Validation can prevent attacks. It uses messages that are not conform to the Web Service description. These attacks are called deviation from message syntax. If we validate the messages that are arriving to the XML, the attacks can be discovered.[22]

5 Practical case: Web Service XML Injection

According to [22], XML Injection tries to modify the XML structure of a SOAP message or other XML document. The injection is performed by inserting content

Proceedings of the Digital Privacy and Security Conference 2022 | 10.11228/dpsc.04.01

like operation parameters, for example, "<" or ">". This attacks are possible if these characters are not escaped appropriately.
An XML Injection attack was performed also on [22]. This attack was performed against a .NET Web Service. This service has two parameters a and b, both int type. The next image shows the SOAP message how the service was invoked.

```
<Envelope>
    <Body>
        <HelloWorld>
            <a> <b>1</b> </a>

            <b> 2 </b>
        </HelloWorld>
    </Body>
</Envelope>
```

Fig. 2. Web Service attack [22]

This message could be the result from an XML Injection attack. 1 was inserted as a parameter content without escaping "<" and ">". This should not be accepted, because it violates the Web Service schema, but in the .NET accepted this message. The resultant parameter values were a=1 and b=0. Thus the attacker was able to modify the value of b just modifying the content of a. With that presented, it is easy to think about how many scenarios are capable of being created and how much restricted data the hacker can access if the changes are bigger than the ones made.

To detect these attacks is to validate the schema on the SOAP message, and if possible include also data validation. If these methods were implemented, the example shown above would not be possible to execute.

6 Conclusion

Along this paper we present numerous issues and threats and some methods to prevent them. These problems and threats are the result of being able to circumvent some implemented measures as well as being able to explore existing vulnerabilities.

With this being said, companies using web services and users are now aware of the security risks while they are using it. Nevertheless, there is still a lot to be done, such as implementing better security measures or hiring white hackers to discover vulnerabilities in web services. In the case of mobiles, the verification of applications made available to consumers, by a person, rather than just a

system. With the increasing demand for web services, new mitigation methodologies and studies are underway, proving that the security area is one of the most important.

References

1. Vaidya, Alpana & Pathak, Vinayak & Vaidya, Ajay. (2016). Mobile Phone Usage among Youth. International Journal of Applied Research and Studies. https://doi.org/10.20908/ijars.v5i3.9483
2. Wang, Yingxu. (2004). Operating Systems. https://doi.org/10.1201/9781420039870.ch144
3. Gilski, P., & Stefanski, J. (2015). Android os: a review. Tem Journal, 4(1), 116.
4. Verma, Nishkarsh & Sambhav, Saurabh. (2020). Development of iOS: A Revolutionary Transformation and the Future. INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING & TECHNOLOGY. https://doi.org/10.34218/IJARET.11.6.2020.040
5. Tasril, Virdyra & Ginting, Meiliyani & Mardiana, Mardiana & Siahaan, Andysah Putera Utama. (2017). Threats of Computer System and its Prevention. International Journal of Scientific Research in Science and Technology.
6. SophosLabs 2019 Threat Report https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-2019-threat-report.pdf. Last accessed 16 December 2021
7. OWASP, https://owasp.org/www-project-top-ten/. Last accessed 4 December 2021
8. OWASP https://owasp.org/Top10/A01_2021-Broken_Access_Control/. Last accessed 7 December 2021
9. OWASP https://owasp.org/Top10/A02_2021-Cryptographic_Failures/. Last accessed 7 December 2021
10. OWASP https://owasp.org/Top10/A03_2021-Injection/. Last accessed 7 December 2021
11. OWASP https://owasp.org/Top10/A05_2021-Security_Misconfiguration/. Last accessed 7 December 2021
12. OWASP https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/. Last accessed 7 December 2021
13. OWASP https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/. Last accessed 7 December 2021
14. Sarhan, Qusay & Gawdan, Idrees. (2018). Web Applications and Web Services: A Comparative Study. Science Journal of University of Zakho. https://doi.org/10.25271/2018.6.1.375
15. W3C, https://www.w3.org/standards/xml/core. Last accessed 4 December 2021
16. Cavanaugh, E. (2006). Web services: Benefits, challenges, and a unique, visual development solution. white paper, Feb, 10.
17. Aruna S, 2016, Security in Web Services- Issues and Challenges, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 05, Issue 09 (September 2016) https://doi.org/10.17577/IJERTV5IS090245
18. IBM, https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=standards-universal-description-discovery-integration-uddi. Last accessed 4 December 2021
19. Instituto Superior Técnico Universidade Lisboa, http://disciplinas.tecnico.ulisboa.pt/leic-sod/2017-2018/labs/05-ws/wsdl/index.html. Last accessed 4 December 2021

20. Nagpal, Bharti & Professor, Asstt & Chauhan, Naresh & Singh, Nanhay & Sharma, Pratima. (2019). Preventive Measures for Securing Web Applications Using Broken Authentication and Session Management Attacks: A Study.
21. Patel, Neel & Shah, Yash & Patel, Neil & Doshi, Manan. (2017). A Review on Prevention for Session Hijacking using One-Time Cookie.
22. Jensen, Meiko & Gruschka, Nils & Herkenhöner, Ralph. (2009). A survey of attacks on web services. Computer Science - R&D. 24. 185-197. https://doi.org/10.1007/s00450-009-0092-6