Web Services Pen-testing Framework for Cyber Security : A Review

Yash Patel[*1], **Dr. Ravi Sheth**[2]
[*1]Research Scholar, MTECH Department, Raksha Shakti University, Ahmedabad, Gujarat, India
[2]Assistant Professor, MTECH Department, Raksha Shakti University, Ahmedabad, Gujarat, India

ABSTRACT

Every single day highly trained Hackers breach the security & take advantage of vulnerabilities to access the confidential and sensitive data. To overcome such problem, the first solution was suggested named Vulnerability Assessment and Penetration Testing (VAPT). However, Penetration testing is done for security holes identification. This paper gives an overview of the stages of penetration testing in a web application for web services. In web services pen-testing, generally, we test for attacks like Web services Foot-printing Attack, Probing Attack, XML Poisoning, and SOAP Injection.

Keywords: Web Application, Penetration Testing, Web Services, XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), XPath (XML Path Language).

I. INTRODUCTION

Web applications provide an interface between end users and servers through a set of web pages that generated at the server end or contain script code to execute dynamically within the client web browser. The Business impact depends on the protection needs of all affected application and data. [2]

Attackers can exploit vulnerable XML processors if they can upload XML or include hostile content in an XML document, exploiting vulnerable code, dependencies or integrations. Attacker's footprint a web application to get a UDDI information such as business entity, business services, binding template and the model. Attackers insert malicious XML code in SOAP requests to perform XML node manipulation or XML schema poisoning in order to generate errors in XML parsing logic and break execution logic. Can manipulate XML external entity references that may lead to arbitrary file or TCP connection openings and may exploit for other web services attacks. XML poisoning enables attackers to cause a Denial-of-service attack and compromise confidential/sensitive information. Moreover, steps for penetration testing taken for web services attacks.[3]
In most cases, web applications communicate with web services (SOAP and RESTful). The former act as a front-end to the latter, which contain the business logic. A hacker might not have direct access to those web services (e.g., they are not on public networks), but can still provide malicious inputs to the web application, thus potentially compromising related services. Typical examples are XML injection attacks that target SOAP communications.[4]

Web services received significant attention recently and several important web service platforms such as .NET are now available. The testing and evaluation of web services are important for both service providers and subscribers.[5]

One way to describe web services is that the components wrapped with SOAP interfaces so they can exchange XML-based messages. This description is simple and reasonably accurate, but it masks some of the complexities. To consider their complexities, we need to consider how traditional programs become web services. Aoyama describes three evolutionary ways [6].

In each, web services often used to publish traditional software on the Internet or to integrate subsystems within an organization. Web services are more widely distributed than traditional software. The fundamental objective of using web services today is the same as that of distributed computing technologies 20 years ago: to allow applications to work cooperatively with other applications over a common network [7]. However, these three methods of software evolution highlight some differences between web services and traditional software.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.[8]

II. FRAMEWORK FOR WEB SERVICES PEN-TESTING

Phase I: Test for XML Structure

- ✓ To Create Structured XML Documents to Build a Denial-of-service attack by overloading the XML parser.
- ✓ Send a large or malformed XML Message to the server.
- ✓ By checking all the parameters being validated, such as: Enumeration, Fractiondigits, Length, Maxexclusive, Maxinclusive, Maxlength MinExclusive, MinInclusive, Minlength, Pattern, Totaldigits, Whitespace.[1]

Phase II: Test for XML Content level

- ✓ Test the web service definition language with the Webscarab tool.
- ✓ Modify the parameter's data based on the WSDL's definition for the parameter.
- ✓ Check whether you can use the web service by escalated privileges.

Phase III: Test for WS HTTP GET Parameters/REST Attacks

Testing HTTP GET query string

https://www.website.com/accountinfo?accountnumber=1234567&userId=aci9485jfuhe92

Result :

<?xml version="1.0">
encoding="ISO-8859-1"?>
<account ="1234567">
<balance>$100</balance>
<body>Bank of targetwebsite account info </body></account>

Phase IV: Test for Suspicious SOAP Attachments

- ✓ Search the Web service definition language (WSDL) which accepts attachments
- ✓ Attach and post a SOAP message with a non-destructive virus such as EICA virus.
- ✓ Set Parameter 'true' in the SOAP response with the Upload File Result, which varies with each service.
- ✓ Store the EICAR test virus file on the host's server and redistribute it as a PDF

Phase V: Test for XPath Injection

- ✓ -XPATH injection is an attack technique used to exploit websites that construct XPath queries from the user-supplied input.
- ✓ -XPATH 1.0 is a language used to refer to parts of an XML Document.
- ✓ -It used directly by an application to query an XML document, or as part of a larger operation such as XSLT Transformation to an XML document, or applying an XQuery to an XML document
- ✓ -The syntax of XPath bears some resemblance to an SQL query and it is possible to form SQL-like queries on an XML document using XPath.

Phase VI: Test for WS Replay

- ✓ Use WebScarab tool as a proxy to capture the HTTP traffic
- ✓ Using the packets captured by WebScarab, use TCPReply to initiate the reply attack by reposting the packet
- ✓ Resend the original message or change the message to determine the host server.
- ✓ Capture many packets within the estimated time to determine session ID patterns in order to assume a valid session ID for the replay attack.

III. RESULTS

As a Result, we can use manual testing approach for Attacks such as Web services Foot-printing Attack, Probing Attack, XML Poisoning, and SOAP Injection.

IV. CONCLUSION & FUTURE WORK

We conclude that In these review paper, we present the stages in which through pen-testing we test web services step by step, and also test for attacks such as Foot-printing Attack, Probing Attack, XML Poisoning, and SOAP Injection. This Review will help us in developing more secure and efficient Web application to provide the better security to the user data.

By these steps of testing you can test web application's web services and to detect attacks which applicable to web services and it is helpful in manual testing approach in web services testing.

V. REFERENCES

[1]. EC-Council Certified Security Analyst (ECSA) v8 Slides.pdf

[2]. SACHIN UMRA, MANDEEP KAUR & GOVIND KUMAR GUPTA, VULNERABILITY ASSESSMENT

[3]. AND PENETRATION TESTING, International Journal of Computer & Communication Technology ISSN (PRINT): 0975 - 7449, Volume-3, Issue-6, 7, 8, 2012.

[4]. https://www.owasp.org/index.php/Top_10-2017_A4-XML_External_Entities_(XXE)

[5]. Sadeeq Jan, Cu D. Nguyen, Andrea Arcuri, Lionel Briand, A Search-based Testing Approach for XML Injection Vulnerabilities in Web Applications, 10th IEEE International Conference on Software Testing, Verification and Validation.

[6]. Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering (HASE'02)1530-2059/02 $17.00 © 2002 IEEE

[7]. M. Aoyama, S. Weerawarana, H. Maruyama, C. Szyperski,. Sullivan, and D. Lea. Web services engineering: Promises and challenges. In Proceedings of the 24th International Conference on Software Engineering, pages 647–648, Orlando, Florida, May 2002.

[8]. J. Clabby. Web services explained: Solutions and applications for the real world. Pearson Education Inc., 2003

[9]. https://en.wikipedia.org/wiki/XML