Smart Intelligent Computing and Communication Technology
V.D. Ambeth Kumar et al. (Eds.)

doi:10.3233/APC210056

The Solution for XML External Entity Vulnerability in Web Application Security

Prabhakar.M[a,1] and Abdul Raheem Syed[b]
[a]*Associate Professor, School of CSE, REVA University, Bangalore, India*
[b]*Associate Consultant, Capgemini Technologies India, Pvt. Ltd, Bangalore, India*

Abstract. Web application security the basic requirement and follow as a minimum security standard nowadays to develop online applications. Most of the web applications are managing online transaction like transferring money, pay bills online and many more services related to finance. So, all the web transactions are exchanged of static data related to both the bank parties which is used to populate XML document. And the SOAP web services and REST protocol are used for exchange of two heterogeneous applications and the data is parsed by XML parser. XML External Entity attack occurs as XML input weakly parsed data. This attack refers problems like confidential information, denial of service and server side request information. In this paper we explain about XML External Entities, prevention measures, problems and expected solutions.

Keywords. Web Application Security, XML External Entity, Sensitive data exposure, Secure Data Protection

1. Introduction

Web Application Security is the latest trend nowadays. Because today all the services are online means the services access any time and any ware. The clients or users are access the application on their convenient time instead of specific office hours also no need to contact with the staff of the office. So, that every client decides instantly as the client's mood change. Once he decides will pay the money also instantly with this online and payment services. Even every application related queries and questionnaire also available online. This services pay the fee with one of the banking services. Here the user may use one of the banking service like Net Banking, Debit Card, Credit Card and also other baking service like PayTm, PhonePe and GooglePay services and many more. From this there may be exchange of information between either two homogeneous or heterogeneous services like Net banking of two same banks or different banks, PayTm to PhonePe or GooglePay so on. The exchange of information between two banking services using any of the web services. The final outcome is communicating with both banking static data and that is very sensitive and critical information. This data is personal data of the bank which need to maintain very secure. Every bank online applications are using different technology for developing their own banking services. The common data for sharing with other bank applications is mostly

[1] Prabhakar. M, Associate Professor, School of CSE , REVA University Bangalore, India; E-mail: prabhakarm@gmail.com.

in the form of XML format for the exchanging of information with any heterogeneous application. Cyber threats and vulnerabilities risk is explains about different security threats that are vulnerable related to risk of online applications. The over all threat types are as per the OWASP top 10 standards as of 2017 follows below

1. SQL Injection: Injection vulnerability are like Structure Query Language (SQL), NoSQL, Operating System and LDAP happens on vulnerable data passed to by the hacker as small portion of code in the executed query. The attacker added data in the query may change the query output for retrieving the data form the table without valid authorization of database. That the query executes with attackers requested data with all the attributes of the table like account details with password information. Based on this information the attacker steal the user sensitive information of the account.
2. Broken Authentication: This vulnerability are frequently happens due to no proper coding standards. The attackers are passes the vulnerable data and stealing the password of the user accounts. Passwords are compromised for login the user account by attackers using vulnerable data on the application.
3. Sensitive data Exposure: Most of the web applications and related API's are not securing the user and organization private data for example banking sector sensitive data, payment card information, hospital sensitive records and social media application data. The attackers may take the sensitive data also try to modify the sensitive data, data may misuse with wrong data which update, financial information may change from one to another also that information lead to legal problems as it exposure as non secure. Sensitive information might be managed without addition security for example encryption remaining data or transferred information and requires mandatory measures when transmission between the application and program.
4. XML External Entity (XXE): legacy or previous version XML versions calculated and managed external entities inside the XML documents. The XML External entities may be utilized for expose sensitive data inside the XML with respect to URI handler, inside records uses, scanning of inside port executing the code remotely and Denial of Service attacks.
5. Broken Access Controls: This is similar as the authorized users can access the sensitive data of the users or organizations personal information. Attackers are use their vulnerable code for accessing the unauthorized information like clients personal information, accounts of clients, see and update the sensitive data and also change the password with access controls rights.
6. Security Misconfiguration: One of the most normal vulnerable issue is Security misconfiguration. Security Misconfiguration is happening basically because of default setting which are applied, fully configured settings are not completed properly, default user and password updating, frequent basic data giving at the time of configurations. On all the cases of installation and framework settings default and weak information uses for configurations lead to Security Misconfiguration.
7. Cross-Site Scripting XSS: XSS vulnerability happening if application contains non-validated data in the latest web page without validation check are not proper. Updates with new patches on old web pages contains user provided information using API that forms new HTML or scripting file. Using cross-site scripting the hackers are accessing the browser session of the user, websites of military services or forwarding the existing session of the applications to do unauthorized transactions like finance and social media accounts information.

8. Insecure De-serialization: This security risk major problem is to execute the code remotely. That to de-serialization flaws do not lead to remote code execution, happen attacks on basic level, SQL Injection vulnerability and authorization related problems.
9. Using Component with known vulnerabilities**:** Using this vulnerability the basic parts like API's, Frameworks and related modules. Execute with similar authorization privileges in the application without proper validations. Deployed component weak that leads to severe data loss or miss transaction like banking and finance or any sensitive data exposure. As part of weak component and API's lead to severe attack happening.
10. Insufficient Logging and Monitoring: Because of not maintained standards also no secure coding, the log information which is sensitive displayed at the time of execution, user critical data like user-id and password logged and displayed after execution in to the log file that lead to leakage of sensitive data to bout. The storage of information on log files and monitoring the sensitive data in any encrypted format is mandatory. Many types of attacks are happening due to the improper logging of sensitive data that displayed in to the log files [5]. In this paper, we describe, detecting, prevention and solution mechanism of the XML External Entity vulnerability and different types of attacks to access sensitive data using XML parser and weak XML elements definition. Stealing and manipulating data at the time of exchanging of two same or heterogeneous banking or financial applications. Due to this the sensitive data of applications are vulnerable because of weak XML parser. The XML External Entity is the one of the vulnerability risks from the OWASP top ten [12-15].

2. XML External Entity (XXE) Processing

A XXE is one kind of attack instead of specific web application which parses XML input. XXE happens if the input of the XML having the related for the external entities are handled by loosely and not strong arranged XML parser. This type of attack which result opened sensitive information, DoS, SSRF, scanning of the port through the point of view with automated where the parser is found also related framework affects. This type of attack mapped record added the use of external DTDs, the external stylesheet, external schema, so on allow on same.

2.1. XML External Entity Attacks

XML External Entity attacks are also called as XXE Injection is a kind of attack that misuse broadly accessible however frequently utilized characteristics of XML parser. Utilization of XML External Entity attacks, an attacker can harm Denial of Service (DoS) just similar usages nearby, remote substance and services. XML External Entity can be utilized to carry out Server Side Request Forgery (SSRF) get the web application to build requests to the different applications. In few situations, XML External Entity attacks that empower port examining and guide for remote code execution. In-band and out-of-band (OOB-XXE) are the two types of XML External Entity attacks. XML (Extensible Markup Language) is a one of the top and common natural data format for accessing any application. XML is utilized from the XML-RPC, SOAP, REST web services via the XML, HTML and DOCX to image files like SVG, EXIF data. To clarify XML information format, the application required an XML parser.

2.2. XML External Entity Attack process

The attacker has the chance of XXE attack for getting the vulnerability from the outside code injection of attack. Based on the XXE attack, the “SYSTEM” recognize to get the inside information on a framework locating application of the XML parser of PHP. Mainly the XXE attack are the because of XML parser and the internet firewall of the application.

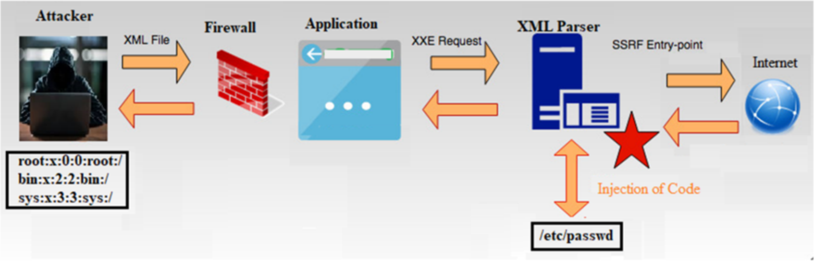

Figure 1. XML External Attack flow

As per the application not required to explicitly responded and gave the result in the attacker related to the doubtful data insecure and viewing without having any encryption or security mechanism. The same attacker can know the Domain Name Service data to break information via sub-domain address to Domain Name Service server which may control by him.

2.3. XML External Entity Risk Factors

XML document parsed by the specified web Application. The executed data is permits inside application code region related to the entity of XML, inside portion of DTD. The DTD is processed and validated by the configured XML processor and also sort out the inside the DTD. The many web sites and blogs are explains similar sample code and expected solutions related to risk factors of XXE.

2.4. Missing XML Validation

The attacker is getting a chance of passing the malicious data as input because of XML parsing validation failure. The attackers understand the weakness of developer’s non-standard insecure code implementation of the application. With the tolerating XML report non validating is the problem instead of Schema or DTD, the developer less knowledge on XML development, is the chance of attacker to pass the malicious, suspicious, malfunction information. This is not feasible for an XML parser to approve data, that means the parser is not suitable for complete syntactic of information. Any how the parser can do the total and intensive activity of validating the archieve information and assurance to the code that forms the record that the information is very much shaped [6].

3. XML External Entity (XXE) Prevention Cheat Sheet

Prevention mechanism of XXE attacks allow a malicious person read the protected record data from the server. The server allows for view the sensitive data form the files in the server first level. Because of not implementing any intrusion detection mechanism and much proper prevention of XXE attacks. Protecting the XML External entities from the attackers based on the language and technology related are as follows Disable the parsing of inline DTD, remove the permissions of the web server which the sensitive data is placed and processed for execution of the user or organization request [9].Some of the sample code based on the technology and language are in the Python use the defused.xml library for XML parsing. Ruby – you can disable expanding of XXE in Nokogiri as updated in many technical blogs and free code reference websites.
XXE injection is one out from OWASP top 10 vulnerability that the attacker passes the malicious input data which is executed by the weak XML parser in the application. This is happening because of the XML elements not in format having related through the external entity executed via the no properly configured XML parser. There are many ways for preventing the XML External Entity attack mainly try to disable entire External Entity of DTD [8]. Based on the parser below method is related for all factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true); Denial Of Service attack secure if we disable DTD parser and external entities must disable on every specified parser.

4. Conclusion and Future Work

In this paper, explains about XML External Entities attacks which may lead to the loss of web application and the organization. The attackers are incorporate uncovered inside data records, that includes private and protected information for example, password, sensitive client information. Some situations XML parser library which is vulnerable to on browser header passed for client-side information. In banking and related organization are exchange of private and sensitive data related to organization using web services. XML External Entity attack, hacker view and modify the XML data inside the web browser which are related to sensitive and private. After the attacker either use the with same credentials or malfunction with malicious data. Most of the attacks are happens with external injection of malicious code or weak of XML Parser, weaker DTD external entities. This paper we explain about major problems of the XXE attack, prevention mechanism and some of the expected solutions to avoid. This is happening mainly because of setting up default configuration done by the developer. The developer required to change the default configuration with valid data and strong parser configuration. And prevention mechanisms of different technologies like Java, PHP, C++ and C#. Future, we will discuss few more XML External Entity detection and prevention mechanisms and security measures at the time of web application development.

References

[1] XML External Entity (XXE) Processing. (2015). Retrieved December 12, 2015, from owasp.org: https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Processing

[2] Facebook Bug Bounty. (2014). Retrieved December 12, 2015, from facebook.com: https://www.facebook.com/BugBounty/posts/778897822124446.

[3] Almroth, Fredrik Nordberg and Karlsson, Mathias. (2014). How we got read access on Google's production servers. Retrieved December 12, 2015 from detectify.com: http://blog.detectify.com/post/82370846588/how-we-got-read-access-on-googlesproduction

[4] Rachel Hogue A Guide to XML External Entity Processing, in Tufts University Comp 116: Introduction to Computer Security Mentor: Ming Chow Fall 2015.

[5] Project Chapters events about OWASP top 10 https://owasp.org/www-project-top-ten/.

[6] Morgan, Timothy D. and Ibrahim, Omar Al. XML Schema, DTD, and Entity Attacks. A Compendium of Known Techniques. (2014). Retrieved December 12, 2015, from vsecurity.com: http://www.vsecurity.com/download/papers/XMLDTDEntityAttacks.pdf

[7] Stuttard, Dafydd. Burp Suite now reports blind XXE injection. (2015). Retrieved December 12, 2015, from portswigger.net: http://blog.portswigger.net/2015/05/burp-suite-now-reports-blind-xxe.html.

[8] Prevention of XML External Entities using Cheat Sheet from OWASP top 10 https://owasp.org/www-project-cheat-sheets/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html.

[9] Prevention of XML External Entities https://www.hacksplaining.com/prevention/xml-external-entities

[10] Mitigte XXE vulnerabilities in Python https://www.acunetix.com/blog/web-security-zone/how-to-mitigate-xxe-vulnerabilities-in-python/ .

[11] Mitigate XXE vulnerabilities in Out-of-band XML External Entity https://www.acunetix.com/blog/articles/band-xml-external-entity-oob-xxe/.

[12] Nanagasabapathy.K et.al. Validation system using smartphone luminescence. IEEE International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Pages: 235 – 239, 6-7 July 2017, Kannur, India

[13] Ambeth Kumar.V.D et.al.Cloud enabled media streaming using Amazon Web Services. IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Pages: 195 – 198, 2-4 Aug. 2017, Vel Tech University, Chennai, India (DOI: 10.1109/icstm.2017.8089150)

[14] Aravindh.B et.al. A novel graphical authentication system for secure banking systems. IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Pages: 177 – 183, 2-4 Aug. 2017, Vel Tech University, Chennai, India

[15] Ambeth Kumar.V.D et.al, (2016).An Efficient Security System for Data base Management from Illegal Access.IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), SSN Engineering College, Chennai, India, 23-25 March, 2016

[16] 25.K. Sabarinathan et.al ., " Machine Maintenance Using Augmented Reality", 3rd International Conference on Communication and Electronics Systems (ICCES), 2018. (DOI: 10.1109/CESYS.2018.8723900)