International Journal of Engineering & Technology, *7 (4.19) (2018) 384-387*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Remote Code Execution in Web Applications

T.Keerthika[1], A.Adithyan[2], S.Balaji[3], Mukulraj J Lunia[4]

[1] *Assistant Professor, Department of IT, Sri Krishna College of Engineering and Technology, Coimbatore.*
[2, 3, 4] *UG Scholar, Department of IT, Sri Krishna College of Engineering and Technology, Coimbatore.*

Abstract

Despite having dedicated applications for different operating system, web application is the most common interface accessed by all the devices. Web application security is an indispensible factor in today's cyber world. Because of the robust resource available on Internet regarding web development, anyone today can develop a website even with zero coding skills. More than developing a perfect website, maintaining the security has become the prime goal today. Huge data breach in companies resulted due to a small security loophole in their website. Even a minor Cross Site Scripting (XSS) bug may lead to the whole server compromise depending upon the attacker who knows how to convert a simple bug into a disaster. Remote Code Execution (RCE) is one of the critical vulnerability that arises due to the unsafe handling of inputs by the server application. This vulnerability arises under various conditions that include but not limited to unsafe deserialization, XML External Entity attack, Server Side Request Forgery and Server Side Template Injection.

Keywords: *Web application, Scripting, Remote Code Execution, Deserialization.*

1. Introduction

All the sensitive data are stored online for accessing the data regardless of the geographical location and for the ease of access. But what developers lag is the security part of the data which is stored online. The developer never thinks in the aspect of the attacker which is the primary reason for the data theft. We are in the information era where each and every bit of information is considered as wealth. Not only passwords are considered as sensitive data. Your email, mobile number, date of birth and every bit of information which your fellow people don't know about you are termed as sensitive. When such data of yours are exposed, there is a high chance of you becoming a prey for Hackers. In recent years, cyber-attacks have been increased in an unbelievable rate especially in India. This is due to the massive growth in the internet users of India. To prevent hacking, Multi-National Companies around the world run bug bounty program where they award bounties for the security researchers who find security loopholes in the Company's application. There are dedicated security researchers whose job is to find security bugs in Applications of the Company which runs bug bounty program. The emergence of Bug bounty program drastically reduced the hack that companies faced. However, still Black Hat hackers tend to hack for money.

2. Literature Survey

In 2012 , Jagnere et al.[1] , discusses the vulnerability of social websites due to the lack of instructions on the RCE, HTML tags, JSP by which user sessions get hacked . It can be more efficient to look for vulnerability instead of finding causes. In 2013, YunhuiZheng [2], proposed a path and context sensitive analysis to detect Remote code execution attacks in web applications. It first creates two abstractions of the program to model the string and non-string behavior, respectively, which are encoded to constraints separately. An algorithm is developed to solve the two types of constrains together. It was very effective in detecting the RCE vulnerabilities in mainly PHP web applications.The analysis reasons about the string and the non-string behavior of a program basically.

In 2013, Animesh Dubey et al [3], proposed an efficient partition technique on web attack detection in the direction of attack time detection in web applications such as (html, php, jsp) and it also includes text documents too. For this to be done the two main factors that are considered primarily was the time factor and the direction of file support. The techniques when compared with traditional one show more effectiveness.

In 2017 ,Ying Dong [4], done a work on malicious code detection in order to prevent the effect of remote code execution. As the work is based on malicious code detection to prevent the code injection attack the mechanism follows the procedure of capturing the HTTP GET request which usually gets logged on the web server. It is usually in the format of CLF (Common Log Format) which consists of the basic information of a HTTP request such as IP address, time stamp, server response status code, web request string, user agent header field and some additional information too. With those information the query is analyzed which is usually identified with "?".This query is then compared with the malicious queries that are already available and used in any previous attacks like SQL, XSS, RCE, XXE etc. Moreover, RCE comes under a special category of Cross Site Scripting attacks can be considered as an exclusive XSS [5].

Shanmuganeethi V [6] had discovered a new method for the detection of Xpath Injection Vulnerabilities in XML database. This XPath Injection is quite similar to SQL injection where the XPath is constructed on the basis of user's input. The attacker can discover how the XML data is structured by crafting a malicious input. To prevent this, Shanmuganeethi proposed a way where the XML document will be checked with the pre-defined values for validness.

Manish Sharma and ShivKumar Singh Tomar [7] had proposed detect and prevent the attacks of Remote Code Execution in 2015. For the prevention of such attacks, they have implemented Reverse Cipher Mechanism and for detection of malicious queries, the made use of Bengin tag along with JSP and HTML.

3.Methodology

Remote Code Execution is a technique where attacker can run arbitrary commands on the target machine. Remote Code Execution (RCE) is application independent and commonly found in all the end points where the user input is sent to the server without sanitization. RCE is a critical flaw where the attacker can take over the whole server if he finds the perfect endpoint. Once he gains the correct end point, the attacker may upload his malicious backdoor to avoid losing the access to the victim machine. RCE has become the most implemented attack against web application where SQL Injection scores the second. As per a survey conducted, RCE accounted for over 96.15 of the web attacks RCE in 2017 [8]. The most common way of achieving a Remote Code Execution is by intercepting the HTTP request and injecting arbitrary commands.

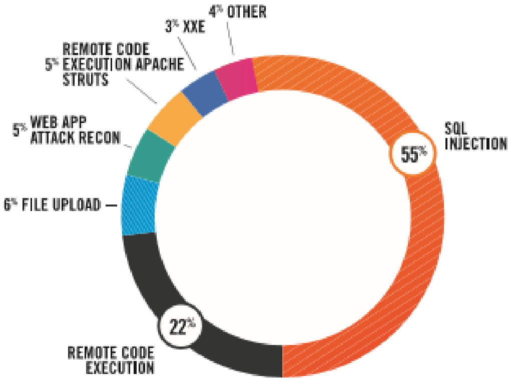


Fig. 1: Web Application Attack Types 2017

According to a research conducted on 2017, RCE has become the second most implemented attack against web application where SQL Injection stands at the first. Whenever you visit a web page using your Browser, a HTTP request is triggered to the server. During its transmission through the layers of the network, the plain text gets converted into binary and other formats according the convenience of the network layers. But finally, the request reaches the server as plain text. Below is an example for a common HTTP request.

```
POST /index.html HTTP/1.1
User-Agent: Mozilla/5.57
Host: www.example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
name=string&content=string&/city=string
```

The above shown is an example for HTTP request with method POST. There are different types of request methods like GET, PUT, HEAD, DELETE, PATCH, OPTIONS.

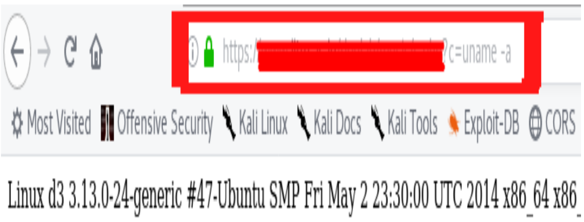
When this HTTP request is sent to the server, the server would send back an HTTP response. The Browser's business is to send HTTP requests, combine the HTTP response received from the server and display the web page. Now going in to the technical part, any user can intercept and manipulate a HTTP request using specialized tools like Burp suite [9], Zed Attack Proxy[10], Fiddler etc. While testing for remote code execution, what the attacker has to look for is the correct parameter where the input is sent to the server without proper sanitization. In the HTTP request shown above, there are three parameters called name, content and city.

These three parameters get filled up by the user's input and sent back to the database. It's sent in plain text without any encryption. This is one of the common mistakes that a developer commits. When passing sensitive information over the POST request, it's a good practice to encrypt the data using at least base64 or MD5. By doing so, even if the input is not sanitized, the attacker cannot achieve remote code execution since the data are sent via encrypted format. But if the attacker has good knowledge on how the server handles the user's input (if he knows the encryption algorithm) the attacker can still attain remote code execution by sending his input in encrypted format. By doing so, while reaching the server, it will get decrypted and eventually, the code requested by the attacker will be executed. A considerable amount of effort has been made to detect malicious queries in web requests using web IDs.

Most of the servers today are Linux based. In older versions of Linux, all the passwords of the users of a Linux machine are stored in etc/passwd file. But since the file was publicly accessible to all the users, the passwords of the users were moved to etc/shadow file. Only administrator has read permissions for that file. Let's consider that our target application does handle the inputs in a unsafe way. To check for RCE, let's manipulate the HTTP request with few basic commands of Linux. Before that, we have to be sure of what header/parameter is handled unsafe by the application. An attacker can even leverage the Content-Length header to execute his code. Let's assume that the attacker is using Directory traversal attack to fetch the etc/passwd file from the Linux root directory. Eventually, while launching the attack, he injects the Content-type header with a Linux command. A manipulated request with malicious code injected will look like this,

```
POST /../../../etc/passwd HTTP/1.1
User-Agent: Mozilla/5.57
Host: www.example.com
Content-Type: XML
Content-Length: uname -a
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
name=string&content=string&/city=string
```

In the above request, the 'Content-Length' parameter is injected with the command 'uname -a'. If the input is not sanitized, the details of the user and the server will be displayed on the response web page. A Remote Code Execution attack can be launched not only by manipulating the HTTP Headers, but also by manipulating the URL. For example, if an parameter is sent via URL and the parameter is handled in a unsecure way, then there exists a possibility of Remote Code Execution.

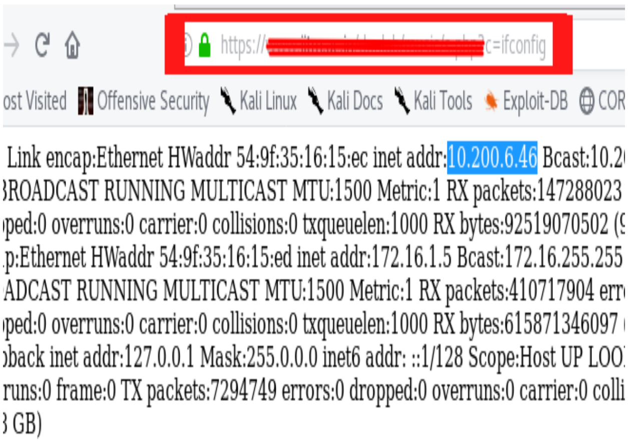

https:/[redacted]?c=uname -a

```
Linux d3 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64 x86_
```

Fig. 2: RCE via URL Parameter

In most cases, the input are sanitized and sent in an encrypted way. This reduces the chance for getting attacked by the Attacker.

But, the problem here is, Encryption doesn't really bother the hacker unless it's encrypted using a private algorithm. But in most cases, Developers prefer Base64 and MD5 because of their convenience. Base64 and MD5 are easier to crack and makes it possible for the attacker to realize the algorithm being implemented in the server. After knowing how the server handles the input, the attacker can encrypt his own malicious payload using the same encryption algorithm and sends it to the server. The payload gets executed when it is decoded by the server.



```
 Link encap:Ethernet HWaddr 54:9f:35:16:15:ec inet addr:10.200.6.46 Bcast:10.2
3ROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:147288023
ped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:92519070502 (9
p:Ethernet HWaddr 54:9f:35:16:15:ed inet addr:172.16.1.5 Bcast:172.16.255.255
ADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:410717904 err
ped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:615871346097
back inet addr:127.0.0.1 Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host UP LOO
runs:0 frame:0 TX packets:7294749 errors:0 dropped:0 overruns:0 carrier:0 colli
3 GB)
```

Fig. 3: LAN Reconnaissance using RCE

On 2017, two highly malicious Remote code execution bug affected all the web servers running on Apache struts2[11]. According to the survey, over 65% of Fortune 100 Companies were running on Apache server with struts2 and all of them were vulnerable to this code execution vulnerability. Struts2 is a framework used in Apache for developing Java Enterprise Edition applications. The vulnerability lies in the java REST Plugin named XStream[12] which is used along with the Struts2 Framework. XStream is actually a java library used for serializing objects into XML. This Library was used right from 2004 and later developed into a plug-in and employed along with Struts2 for serializing and de-serializing any other forms of objects into XML form.

Mostly, java deserialization [13][14] is a technique employed to attain remote code execution in the target system. The weapon delivered to the victim's host will be self-executed and by exploiting the vulnerabilities of the operating system and/or the applications, it installs the malware to keep a door open for theattacker. Generally, in Java, the concept of Serialization and De-Serialization has a robust range of applications and widely used all over the world. In layman terms, serialization is the process where an object is converted into stream of bytes for the purpose of storage or transferring through remote computers. De-serialization is the reverse of serialization where the stream of bytes is reconstructed to form an object. This Serialization and deserialization becomes a severe security vulnerability when the user supplied untreated input is serialized and de-serialized without checking. A Java Deserialization attack can be executed in the following scenario. The Plug-in doesn't check for malicious entry and it de-serializes all the input given by the user in the Content-type header. On receiving a request from the attacker, the plug-in checks for the Content-type header and if it's set to application/xml, the plugin performs unsafe de-serialization without any sanitization. This allowed the attacker to inject malicious code into the crafted HTTP header "Content-type" and while de-serializing the input into XML entity; the malicious input injected by the attacker in the Content-type header gets executed. The vulnerability was widely exploited by the attackers after the POC code was released by the security researchers. The commands executed by the attackers on the Imperva Protected Applications are researched by Imperva and they had released a statistic map based on the commands.

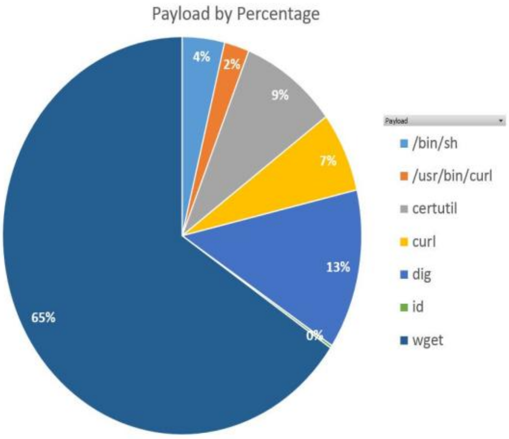


Fig. 4: Commands exploited with Struts2 POC

The vulnerability was assigned with CVE-2017-9805. A month later this bug was released, another remote code execution was discovered in Apache Struts2 Framework. In Struts2, the content type header was configured to throw out an error when invalid input is received. An attacker can leverage this Content-Type header to inject his malicious commands. This is due to the presence of Jakarta Multipart Parser. The vulnerability has been assigned with CVE-2017-5638.

4. Conclusion

Thus, these kinds of Remote Code Execution attacks can be launched against all the applications that handle the inputs in an unsafe manner. Validating inputs is not the only remedy for this kind of Code Execution attacks. Beyond that, usage of cryptographic protocols and enforcing encrypted communications between the client and server and server other techniques along with common security protocols like authentication, access control, and security policy [15][16][17] when employed together will reduce the risk of these attacks while complete eradication of this attack is still a hoax. Installation of Intrusion Detection System (IDS) is an advised measure to mitigate these attacks. We are in a digital era where we have to safe guard our own privacy as well as our data. The best way to mitigate these kinds of attacks is by running a bug bounty program through platforms like HackerOne and BugCrowd. By inviting Hackers to perform penetration test on your application, you can know about the level of your security from a Hacker's perspective.

References

[1] Jagnere, P., "Vulnerabilities in social networking sites," Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on, pp.463, 468, 6-8 Dec. 2012.

[2] Y. Zheng and X. Zhang, "Path sensitive static analysis of web applications for remote code execution vulnerability detection," 2013 35th International Conference on Software Engineering (ICSE), San Francisco, CA, 2013, pp. 652-661.doi: 10.1109/ICSE.2013.6606611

[3] AnimeshDubey, Ravindra Gupta, Gajendra Singh Chandel,An Efficient Partition Technique to reduce the Attack Detection Timewith Web based Text and PDF files , International Journal of Advanced Computer Research (IJACR),Volume-3 Number-1 Issue-9March-2013.

[4] Ying Dong, Yuqing Zhang, "Adaptively Detecting Malicious Queries in Web Attacks".

[5] Y. Zheng, X. Zhang, Path sensitive static analysis of web applicationsfor remote code execution vulnerability detection, in: International Conference on Software Engineering, 2013, pp.

[6] 652–661.

[7] Shanmughaneethi, V., R. Ravichandran, and S. Swamynathan. "PXpathV: Preventing XPath Injection Vulnerabilities in Web

Applications." International Journal on Web Service Computing 2.3, 2011.

[8] Manish Sharma and Shivkumar Singh Tomar “Attack Detection and Security in Remote Code Execution” International Journal of Computer Applications (0975 – 8887) Volume 114 – No. 14, March 2015

[9] J. Fonseca, M. Vieira, H. Madeira, Evaluation of web securitymechanisms using vulnerability & attack injection, Dependable& Secure Computing IEEE Transactions on 11 (5) (2014) 440–453.

[10] R. E. L. de Jiménez, "Pentesting on web applications using ethical - hacking," 2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI), San Jose, 2016, pp. 1-6.doi: 10.1109/CONCAPAN.2016.7942364

[11] Y. Makino and V. Klyuev, "Evaluation of web vulnerability scanners," 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, 2015, pp. 399-402. doi: 10.1109/IDAACS.2015.7340766

[12] N. Qi and Z. Yang, "Research of Struts2 framework and web application based on Ajax," 2009 IEEE International Symposium on IT in Medicine & Education, Jinan, 2009, pp. 903-908.doi: 10.1109/ITIME.2009.5236203

[13] E. Y. C. Wong, A. T. S. Chan and Hong Va Leong, "Xstream: a middleware for streaming XML contents over wireless environments," in IEEE Transactions on Software Engineering, vol. 30, no. 12, pp. 918-935, Dec. 2004. doi: 10.1109/TSE.2004.108

[14] R. C. Seacord, "Java Deserialization Vulnerabilities and Mitigations," 2017 IEEE Cybersecurity Development (SecDev), Cambridge, MA, 2017, pp. 6-7. doi: 10.1109/SecDev.2017.13

[15] R.Maheswari, S.Sheeba Rani, V.Gomathy and P.Sharmila,“Real Time Environment Simulation through Virtual Reality” in International Journal of Engineering and Technology(IJET) , Volume.7, No.7, pp 404-406, April 2018

[16] D. Cappelli, A. Moore, R. Trzeciak, and T. J. Shimeall, “Common sense guide to prevention and detection of insider threats 3rd edition–version 3.1,” Published by CERT, Software Engineering Institute, Carnegie Mellon University, http://www. cert. org, 2009.

[17] R. H. Anderson, “Research and development initiatives focused on preventing, detecting, and responding to insider misuse of critical defences information systems.” RAND CORP SANTA MONICA CA, Tech. Rep., 1999.

[18] J. Hunker and C. W. Probst, “Insiders and insider threats-an overview of definitions and mitigation techniques.”JoWUA, vol. 2, no. 1, pp. 4–27, 2011.

[19] Dipon Kumar Ghosh, Prithwika Banik , Dr. S. Balakrishnan (2018), Review-Guppy: A Decision-Making Engine for Ecommerce Products Based on Sentiments of Consumer Reviews”, International Journal of Pure and Applied Mathematics, Volume 119, No. 12, 2018, pp.1135-1141.

[20] Venkatachalam K, S.Balakrishnan, R.Prabha, S.P.Premnath, Effective Feature Set Selection And Centroid Classifier Algorithm For Web Services Discovery”, International Journal of Pure and Applied Mathematics, Volume 119, No. 12, 2018, pp.1157-1172.

[21] S. Balakrishnan, A. Jebaraj Rathnakumar and K. N. Sivabalan, “Information Security in D-Media (Digital Media)”, ARPN Journal of Engineering and Applied Sciences. May 2016, Vol. 11, No. 9, pp. 5707- 5710.