IT22199508

SLIIT

Discover Your Future

Sri Lanka Institute of Information Technology

Bug Bounty Report 03

mattermost

IE2062 – Web Security

Submitted by:

IT22199508 – Athapaththu A.M.M.I.P

Date of submission

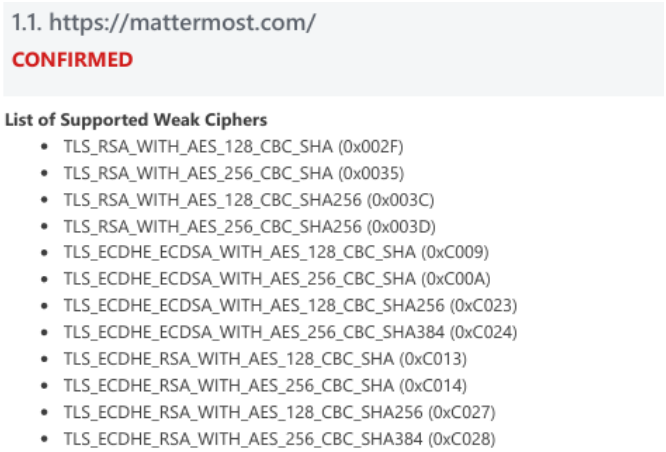
2024.05.04

Table of Contents

IT22199508

Vulnerability

1) Weak Ciphers Enabled


1.1. https://mattermost.com/

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003D)
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xC009)
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xC00A)
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xC023)
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xC024)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)

Date of Discovery: 20/04/2024

Date of Reporting: 21/04/2024

I am writing to inform you of a vulnerability that I have found in your website, http://mattermost.com/ . (A02) 2021: Cryptographic Failures shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often comes up to sensitive data exposure or system compromise. This may lead to several risks of the website. You can see the detailed report of this vulnerability below.

- ## Description

When activated, weak ciphers provide a serious risk to the safety of data and computer systems. These weak encryption methods give a gap that can be exploited by hostile actors since they are not complicated enough or strong enough to survive present cyberattacks. Attackers can intercept, decode and alter data by weakening the encryption used to secure sensitive information, jeopardizing the communication's secrecy and integrity. Given the importance of data security and privacy in today's linked society, this vulnerability is particularly serious.

- ## Impact Assessment

Under the OWASP category connected to poor Cipher Enabled vulnerabilities within the domain of **http://mattermost.com**/, various potential security threats have been found due to the usage of poor encryption protocols or ciphers. The impact of not addressing this vulnerability is considerable and includes the following potential,

consequences:

1. Data Breaches: The use of weak ciphers raises the risk of data breaches. If attackers successfully infiltrate the website, they may obtain access to a goldmine of sensitive user information, leading to potential data breaches and associated legal and financial penalties.

2. Regulatory Non-Compliance: Using weak ciphers may put Starbucks at variance with data protection rules and industry norms, potentially leading to legal and financial implications for non-compliance.

3. Loss of User Trust: Users expect their data to be handled securely. When weak ciphers are employed, it erodes trust in the website's security. This loss of trust can significantly damage the reputation and credibility of Starbucks and may lead to a reduction in customer confidence.

4. Enhanced Attack Surface: Weak ciphers may offer an enhanced attack surface for cybercriminals, making it easier for them to exploit weaknesses and gain unauthorized access to the website, its servers, and underlying systems.

It is vital to fix this Weak Cipher Enabled issue swiftly by adopting strong encryption methods and ciphers. Failure to do so not only expose users to potential security concerns but also damages Starbucks' reputation, legal compliance, and general corporate integrity. The repercussions of the vulnerability will depend on the actions taken by hostile actors and the level of the website's security flaws.

- ## Request

```
[NETSPARKER] SSL Connection
```

- ## Response

```
Response Time (ms) : 1    Total Bytes Received : 27    Body Length : 0    Is Compressed : No

[NETSPARKER] SSL Connection
```

- ## Actions to take

1. For Apache, you should modify the SSLCipherSuite directive in the httpd.conf

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**
    a.Click Start, click Run, type regedt32or type regedit, and then click OK.
    b.In Registry Editor, locate the following registry key:
HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders
    c.Set "Enabled" DWORD to "0x0" for the following registry keys

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

- ## Remediation

Configure your web server to disallow using weak ciphers.

2) Missing Content-Type Header

Date of Discovery: 20/04/2024

Date of Reporting: 21/04/2024

- ## Description

MIME type sniffing is a standard functionality in browsers to find an appropriate way to render data where the HTTP headers sent by the server are either inconclusive or missing.

This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the intended content type.

The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image

- Request

Request

```
GET /cdn-cgi/ HTTP/1.1
Host: mattermost.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: __cf_bm=om9Su8PHIosebxiNoh4YVadPB.wlmU6ad2gGwaBZ_M8-1715282678-1.0.1.1-kCrKV8HFfIp3kXaO23A1FRhV
vV1lOg1CxMCzqdk639OU0OHe6.2G4RNXD3ftT2h.mE_FuS1ysex8O3QlTtIh7g
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

- Response

Response

Response Time (ms) : 401.0223 **Total Bytes Received** : 195 **Body Length** : 0 **Is Compressed** : No

```
HTTP/1.1 404 Not Found
alt-svc: h3=":443"; ma=86400
Server: cloudflare
CF-RAY: 881412247ea37245-CMB
Connection: keep-alive
Transfer-Encoding: chunked
Date: Thu, 09 May 2024 19:24:38 GMT
```

- ## Remediation

1. When serving resources, make sure you send the content type header to appropriately match the type of the resource being served. For example, if you are serving an HTML page, you should send the HTTP header.

```
Content-Type: text/html
```

2. Add the X Content Type Options header with a value of **"nosniff"** to inform the browser to trust what the site has sent is the appropriate content type, and to not attempt **"sniffing"** the real content type.

```
X-Content-Type-Options: nosniff
```

IT22199508