



Penetration Testing Report

for

KSC .NET 6 Upgrade

Report Number: 202306002

Version: 0.1

Responsible: Nipon Taikham (IT Security Controller and Advisory)

Document Class: Confidential

Document Details

Document Usage

This document contains confidential and proprietary information.

It is intended for internal use of **Ayudhya Capital Services Company Limited (AYCAP)**.

Unauthorized use or disclosure of reproduction of this document is prohibited.

Version History

Version	Date (DD-MM-YYYY)	Author	Remark
0.1	15-06-2023	Nipon Taikham	Draft report

Table of Contents

Document Details	2
Document Usage.....	2
Version History.....	2
1. Executive Summary	4
1.1 Executive Overview.....	4
1.2 Summary Results.....	4
2. Technical Summary	4
2.1 Technical Summary.....	4
2.2 Technical Impact.....	5
2.3 Scope of Work.....	5
2.4 Technical Risk Assessment.....	5
2.5 Table of Findings.....	6
3. Detailed Analysis	7
3.1 Information Disclosure via HTTP Banners.....	7
3.2 Vulnerable JavaScript Dependency.....	11
3.3 Cookie Without Secure Flag.....	14
3.4 Missing Security Headers.....	17
4. Appendix	21

1. Executive Summary

1.1 Executive Overview

The core component named .NET 6, which is working as a core system of the Krungsriconsumer.com website was upgraded to the newer version to solve multiple issues. Since the system was enhanced with the core upgrade, the system is required to ensure that there is no severe vulnerability found including security-related configuration by the proof of security assessments. This report is a penetration testing report developed to expose all findings to related parties and those core-upgrade-related vulnerabilities must be ultimately mitigated before go-live.

1.2 Summary Results

During the assessment, the tester discovered **4 low-severity issues** in the system. By the way, the findings are not results from the upgrade of the component but from another layer of the system. So, the findings will be treated as BAU issues which will not block the go-live process of the application.

In the worst case scenario, the most notable vulnerability could result in the following consequences:

- An attacker intercepts and read data from insecure configuration of the HTTP banner and security headers.
- An attacker performs vulnerabilities in outdated software to cause information disclosure and service interruptions.
- An attacker may gather useful information from insecure configuration which could lead to further attack in the future.

2. Technical Summary

2.1 Technical Summary

There are 4 low-severity issues found during the assessment. All of them are not related to the .NET upgrade components but one of them is application related issues.

The findings named “Vulnerable JavaScript Dependency” is the application-related issue which requires application team to plan and upgrade to the newer supported version. The upgrade of these dependencies requires testing in both regression test and unit test to ensure that all functionalities are working properly.

The remaining three issues relate to middleware configuration. A list of HTTP banners that contain server’s information like IIS version, ASP.NET in used are required to remove from the HTTP response. The ‘Secure’ flag for all session cookies is required to be set. And multiple security headers especially specified within the IT Security Requirements are needed to be properly configured.

2.2 Technical Impact

The summary of the major vulnerabilities is as follows:

1. General Information Disclosure

The system running the application was configured with insecure practice or missing required components such as HSTS header that enforces the application to only support secure channel for sensitive data transmission.

2. Potential Cross-Site Scripting Attacks

The list of third-party libraries is being used by the application. Multiple Cross-Site Scripting from the outdated version may be exploited and results in session hijack or identity thief.

2.3 Scope of Work

The scope of penetration test for KSC .NET 6 Upgrade project is to discover all kinds of security vulnerabilities and security misconfigurations in .NET 6 core components in the real implementation of the application. This assessment was performed with total effort of 2 man-days without source code review. The targets below are considered as targets in scope.

- 1) Uat.krungsriconsumer.com
- 2) Non-prod.krungsriconsumer.com

Testing Approach: Grey-box without credential. The tester was only provided with firewall connectivity to the targets.

Testing environment: UAT

2.4 Technical Risk Assessment

The risk calculation uses 5 X 5 matrix which the result of the risk was calculated from the multiplication of likelihood level and impact level. If there is no likelihood or no impact, the rating 'information' may be used.

Risk Formula	
Risk Level = Likelihood x Impact	

Risk Criteria and Level of Risk						
Risk Value		Likelihood Level				
		1	2	3	4	5
Impact Level	5	High (5)	High (10)	Critical (15)	Critical (20)	Urgent/Emergency (25)
	4	Medium (4)	High (8)	High (12)	Critical (16)	Critical (20)

	3	Low (3)	Medium (6)	High (9)	High (12)	Critical (15)
	2	Low (2)	Low (4)	Medium (6)	High (8)	High (10)
	1	Very Low (1)	Low (2)	Low (3)	Medium (4)	High (5)

CVSS Score
CVSS:3.1/

The CVSS score is calculated by using the official website from NVD. For more information, please see the link below.

[NVD - CVSS v3 Calculator \(nist.gov\)](https://nvd.nist.gov/vuln/calculator)

2.5 Table of Findings

No.	Finding Name	Affected System/URL	Risk	Page
1	Information Disclosure via HTTP Banners	https://uat.krungsriconsumer.com https://non-prod.krungsriconsumer.com	Low	7
2	Vulnerable JavaScript Dependency	https://non-prod.krungsriconsumer.com	Low	11
3	Cookie Without Secure Flag	https://non-prod.krungsriconsumer.com	Low	14
4	Missing Security Headers	https://uat.krungsriconsumer.com	Low	17

3. Detailed Analysis

3.1 Information Disclosure via HTTP Banners

3.1.1 Description

The server's information such as Microsoft IIS version, specific name of technology in use, and other server-related information can be retrieved by simply observing HTTP responses. Successful attacks of these findings help an attacker to gather useful information which ultimately exploit in later stage of their attacks.

Affected Hosts:

```
https://uat.krungsriconsumer.com  
https://non-prod.krungsriconsumer.com
```

3.1.2 Steps to Reproduce

By using the inspect mode of any web browser or, in this case, web proxy interception, an attacker can gather information of the targets.

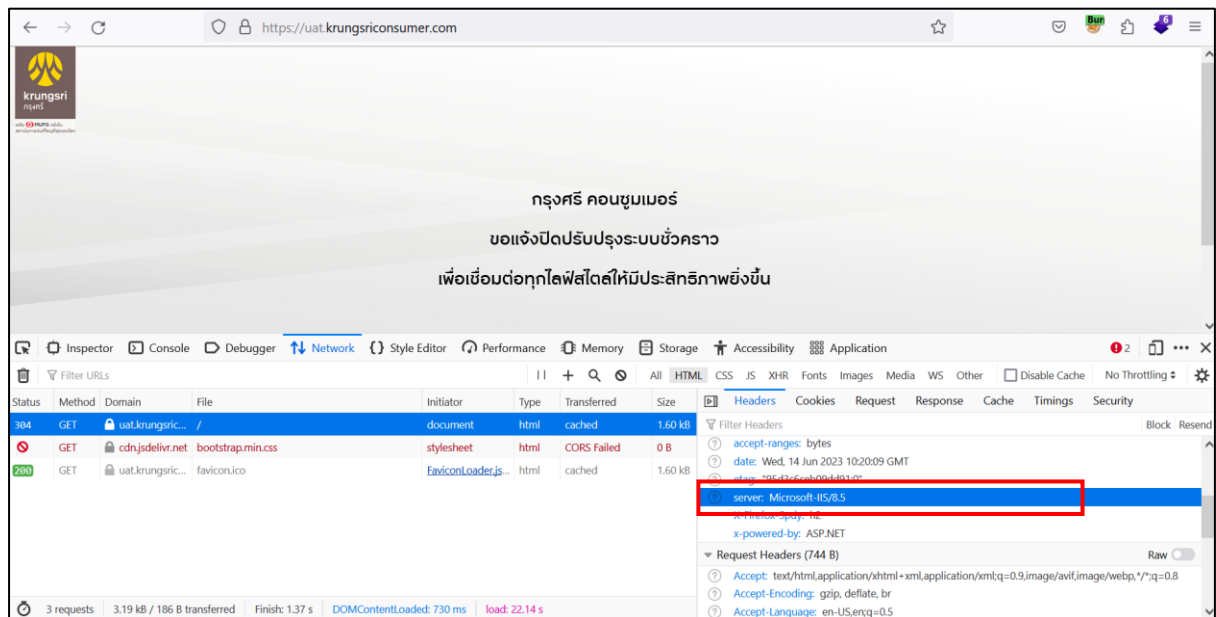
- 1) <https://uat.krungsriconsumer.com>

When accessing the URL, "https://uat.krungsriconsumer.com", below HTTP response can be gathered from web proxy interception tool. Please note that the version of IIS can be found from the HTTP response.



```
Response  
Pretty Raw Hex Render  
1 HTTP/1.1 200 OK  
2 Content-Type: text/html  
3 Last-Modified: Tue, 13 Jun 2023 04:37:49 GMT  
4 Accept-Ranges: bytes  
5 ETag: "95d3c6ceb09dd91:0"  
6 Vary: Accept-Encoding  
7 Server: Microsoft-IIS/8.5  
8 X-Powered-By: ASP.NET  
9 Date: Wed, 14 Jun 2023 10:09:29 GMT  
10 Connection: close  
11 Content-Length: 1597  
12  
13 <!doctype html>  
14 <html lang="en">
```

Even using the 'Inspect' mode for the Firefox browser or similar browser, the HTTP response can be found as shown below.

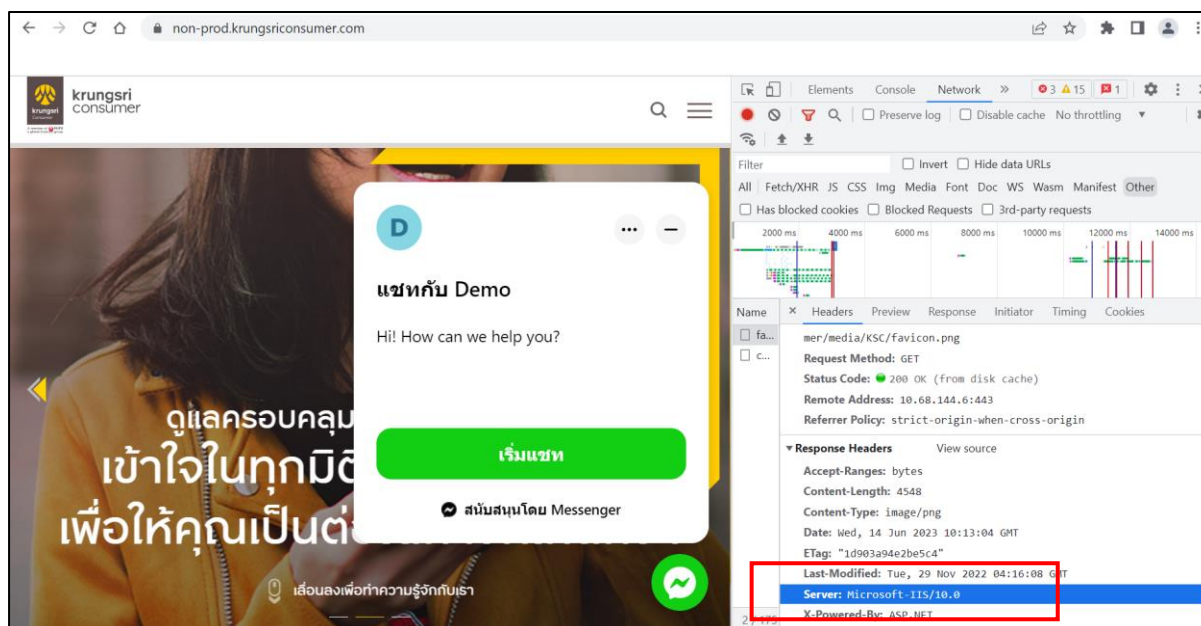


2) <https://non-prod.krungsriconsumer.com>

The HTTP response from the URL, “<https://non-prod.krungsriconsumer.com>”, shows that the header “Server: Microsoft-IIS/10.0” discloses the version of the IIS and “X-Powered-By: ASP.NET” discloses the web technology in use.



Using the “Inspect” mode from Chrome web browser also shows the information as shown in the screenshot below.



3.1.3 Solution/Mitigation

It is recommended to remove the server’s information from the HTTP response to prevent unnecessary headers in the production environment.

For the IIS, these steps can be used to mitigate the findings.

1) X-Powered-By Header

The HTTP header "X-Powered-By" reveals the version of IIS being used on the server. This can be disabled by:

- 1.1 Open the IIS Manager.
- 1.2 Select the website that Secret Server is running under.
- 1.3 Select "HTTP Response Headers".
- 1.4 Select the "X-Powered-By" HTTP Header and select "Remove".

2) Server Header

- 2.1 Open the IIS Manager.
- 2.2 Select the website that Secret Server is running under.
- 2.3 Select "Configuration Editor".
- 2.4 Navigate to "system.webServer/security/requestFiltering" then set the value of the key "removeServerVariable" to "True".

For more information, please see this link.

<https://techcommunity.microsoft.com/t5/iis-support-blog/remove-unwanted-http-response-headers/ba-p/369710>

3.1.4 Risk Calculation

Overall Risk: Low

Risk Formula
Risk Level = Likelihood x Impact

Risk Criteria and Level of Risk						
Risk Value		Likelihood Level				
		1	2	3	4	5
Impact Level	5	High (5)	High (10)	Critical (15)	Critical (20)	Critical (25)
	4	Medium (4)	High (8)	High (12)	Critical (16)	Critical (20)
	3	Low (3)	Medium (6)	High (9)	High (12)	Critical (15)
	2	Low (2)	Low (4)	Medium (6)	High (8)	High (10)
	1	Low (1)	Low (2)	Low (3)	Medium (4)	High (5)

CVSS Score
CVSS:3.1 N/A

3.2 Vulnerable JavaScript Dependency

3.2.1 Description

The vulnerable version of dependencies which are bootstrap and jQuery in use. The use of third-party JavaScript libraries can introduce a range of DOM-based vulnerabilities. Successful attacks of these findings help an attacker to hijack user account like the exploitation of DOM-based Cross Site Scripting.

Affected Hosts:

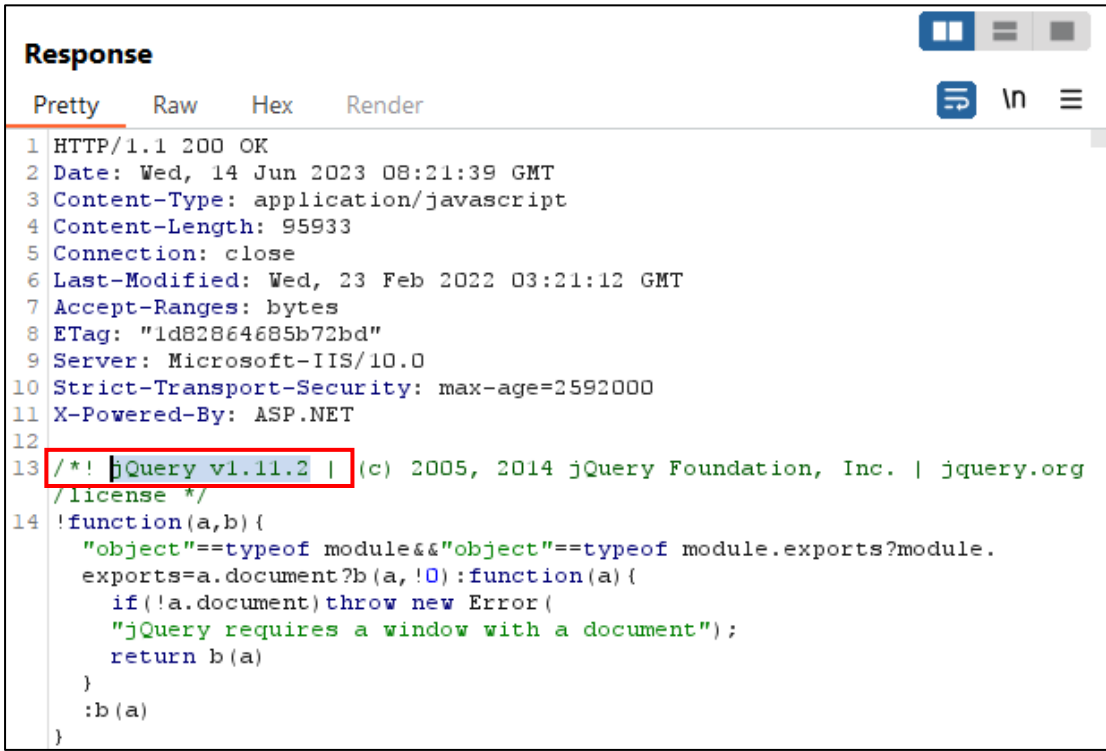
```
https://non-prod.krungsriconsumer.com
```

3.2.2 Steps to Reproduce

When accessing some well-known files for the JavaScript libraries such as “jquery.min.js”, and “bootstrap.min.js”, returns the version of the libraries in use.

For example, by accessing this URL then using a web proxy interception tool to observe the result, the following screenshot can be obtained.

- 1) URL: <https://non-prod.krungsriconsumer.com/js/ksc/jquery.min.js>



```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Wed, 14 Jun 2023 08:21:39 GMT
3 Content-Type: application/javascript
4 Content-Length: 95933
5 Connection: close
6 Last-Modified: Wed, 23 Feb 2022 03:21:12 GMT
7 Accept-Ranges: bytes
8 ETag: "1d82864685b72bd"
9 Server: Microsoft-IIS/10.0
10 Strict-Transport-Security: max-age=2592000
11 X-Powered-By: ASP.NET
12
13 /*! jQuery v1.11.2 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.org
/license */
14 !function(a,b){
    "object"===typeof module&&"object"===typeof module.exports?module.
    exports=a.document?b(a,!0):function(a){
        if(!a.document)throw new Error(
            "jQuery requires a window with a document");
        return b(a)
    }
    :b(a)
}
```

The above screenshot shows that the jQuery version 1.11.2 which contains at least a Cross-Site Scripting vulnerability is in use within the application.

- 2) URL: <https://non-prod.krungsriconsumer.com/js/ksc/bootstrap.min.js>

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Wed, 14 Jun 2023 08:24:14 GMT
3 Content-Type: application/javascript
4 Content-Length: 48952
5 Connection: close
6 Last-Modified: Wed, 23 Feb 2022 03:21:12 GMT
7 Accept-Ranges: bytes
8 ETag: "1d82864685abb38"
9 Server: Microsoft-IIS/10.0
10 Strict-Transport-Security: max-age=2592000
11 X-Powered-By: ASP.NET
12
13 /*!
14 * Bootstrap v4.0.0 (https://getbootstrap.com)
15 * Copyright 2011-2018 The Bootstrap Authors (
16   https://github.com/twbs/bootstrap/graphs/contributors)
17 * Licensed under MIT (
18   https://github.com/twbs/bootstrap/blob/master/LICENSE)
19 */
20 !function(t,e){
21   "object"==typeof exports&&"undefined"!=typeof module?e(exports,require
22     ("jquery"),require("popper.js")):"function"==typeof define&&define.amd
23     ?define(["exports","jquery","popper.js"],e):e(t.bootstrap={
24     },
25     t.jQuery,t.Popper)
26 }
    
```

The above result indicates that the Bootstrap version 4.0.0. This version of the library contains three Cross-Site Scripting vulnerabilities.

3.2.3 Solution/Mitigation

It is recommended to upgrade the libraries to be the latest version, especially on the production environment.

For .Net environment, upgrading the packages via NuGet Package Manager is recommended.

For more information, please see this link.

[NuGet Gallery | bootstrap 5.3.0](#)

[NuGet Gallery | jQuery 3.7.0](#)

3.2.4 Risk Calculation

Overall Risk: Low

Risk Formula	
Risk Level = Likelihood x Impact	

Risk Criteria and Level of Risk					
Risk Value	Likelihood Level				
	1	2	3	4	5

Impact Level	5	High (5)	High (10)	Critical (15)	Critical (20)	Critical (25)
	4	Medium (4)	High (8)	High (12)	Critical (16)	Critical (20)
	3	Low (3)	Medium (6)	High (9)	High (12)	Critical (15)
	2	Low (2)	Low (4)	Medium (6)	High (8)	High (10)
	1	Low (1)	Low (2)	Low (3)	Medium (4)	High (5)

CVSS Score
CVSS:3.1 N/A

3.3 Cookie Without Secure Flag

3.3.1 Description

The 'Secure' flag is set to enforce that the browser will not submit the cookie in any requests that are used an unencrypted HTTP connection. The Secure flag should be set on all cookies that are used for transmitting sensitive data over HTTPS channel to prevent cookie-hijacking-related issues.

Affected Cookies:

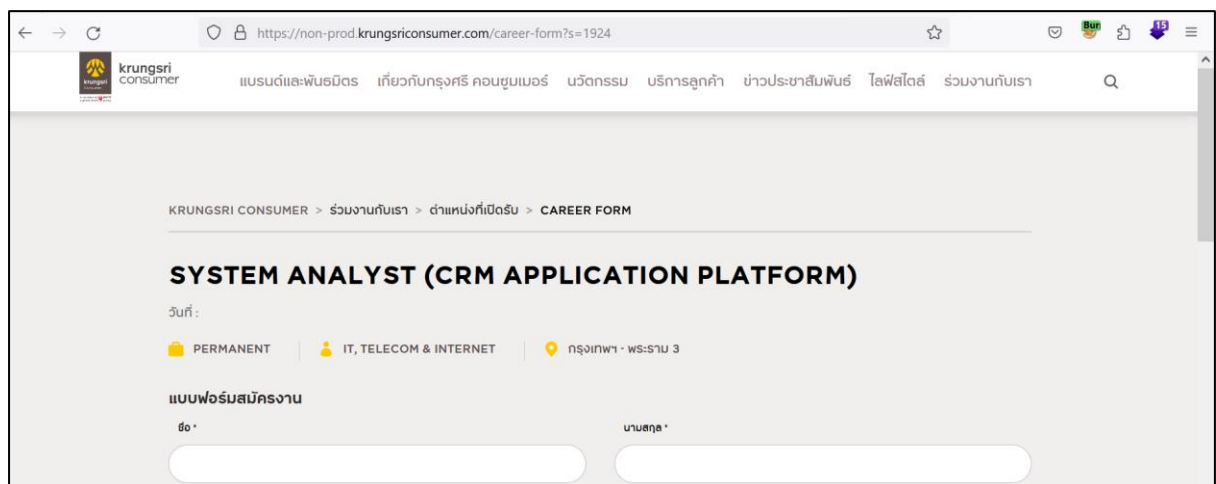
```
.AspNetCore.Antiforgery.zRiataXIOAs  
.AspNetCore.Mvc.CookieTempDataProvider
```

Affected Hosts:

```
https://non-prod.krungsriconsumer.com
```

3.3.2 Steps to Reproduce

URL: <https://non-prod.krungsriconsumer.com/career-form?s=1924>



In this sample, when accessing a page "Career Form" of the website, the HTTP returns with a new pair of cookies being set from the server side. By the way, an attacker can be observed from the HTTP response that the two cookies are not properly set with 'Secure' flag.

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Thu, 15 Jun 2023 02:08:18 GMT
3 Content-Type: text/html; charset=utf-8
4 Connection: close
5 Cache-Control: no-cache, no-store
6 Pragma: no-cache
7 Server: Microsoft-IIS/10.0
8 Strict-Transport-Security: max-age=2592000
9 Set-Cookie: .AspNetCore.Antiforgery.zRiataXIOAs=
  CfDJ8ByUdr8_gBpCuGNOGLjwHFfmAd4wA-XOdgmQgWYkprWiZR_BnW5vDJ6LMJc-tNMge4p
  rflz2nH669nF9FGGVLzdKfqeys-sr_KwfpCmDohh9u18JGEOAWHx1nn_jtBz7vUQncy2TqP
  IvPGwOHVE_r4o; path=/; samesite=strict; httponly
10 Set-Cookie: .AspNetCore.Mvc.CookieTempDataProvider=
  CfDJ8ByUdr8_gBpCuGNOGLjwHFd5hZJe8uvGQbttEnhYaUttgA_rkInKNZnqlzWQiTtyf_J
  CURT_24XH68gsit7oaQu-KGzCwI6gWXIH-FcN3xgi7G1LERdLKOz07TkvVbswhYCHn9_yoK
  MKsQCvK1jPD94sUT3hMNA4NbtuaKHbXb12; path=/; samesite=lax; httponly
11 X-Frame-Options: SAMEORIGIN
12 X-Powered-By: ASP.NET
13 Content-Length: 34896
14
15 <!DOCTYPE html>
16 <html lang="en">

```

The above result indicates that the Bootstrap version 4.0.0. This version of the library contains three Cross-Site Scripting vulnerabilities.

3.3.3 Solution/Mitigation

The secure flag should be set on all cookies that are used for transmitting sensitive data when accessing content over HTTPS. If cookies are used to transmit session tokens, then areas of the application that are accessed over HTTPS should employ their own session handling mechanism, and the session tokens used should never be transmitted over unencrypted communications.

To enable the 'Secure' flag in IIS, edit the web.config file for the URL Rewrite method.

```

<system.webServer>
  <rewrite>
    <outboundRules>
      <rule name="Use only secure cookies" preCondition="Unsecured cookie">
        <match serverVariable="RESPONSE_SET_COOKIE" pattern=".*" negate="false" />
        <action type="Rewrite" value="{R:0}; secure" />
      </rule>
    <preConditions>
      <preCondition name="Unsecured cookie">
        <add input="{RESPONSE_SET_COOKIE}" pattern="." />
      </preCondition>
    </preConditions>
  </rewrite>
</system.webServer>

```

```

        <add input="{RESPONSE_SET_COOKIE}" pattern=""; secure" negate="true" />
    </preCondition>
</preConditions>
</outboundRules>
</rewrite>
...
</system.webServer>
    
```

For more information, please see this link.

<https://cwe.mitre.org/data/definitions/614.html>

[How to Setting the Secure and HTTPOnly flags on the JSESSIONID cookie in IIS? - Microsoft Q&A](#)

3.3.4 Risk Calculation

Overall Risk: Low

Risk Formula	
Risk Level = Likelihood x Impact	

Risk Criteria and Level of Risk						
Risk Value		Likelihood Level				
		1	2	3	4	5
Impact Level	5	High (5)	High (10)	Critical (15)	Critical (20)	Critical (25)
	4	Medium (4)	High (8)	High (12)	Critical (16)	Critical (20)
	3	Low (3)	Medium (6)	High (9)	High (12)	Critical (15)
	2	Low (2)	Low (4)	Medium (6)	High (8)	High (10)
	1	Low (1)	Low (2)	Low (3)	Medium (4)	High (5)

CVSS Score
CVSS:3.1 N/A

3.4 Missing Security Headers

3.4.1 Description

The list of required security headers is missing from the HTTP response such as Strict-Transport-Security, Content-Security-Policy, X-Frame-Options, X-XSS-Protection, Cache-Control, and X-Content-Type-Options. These cookies are serving different purposes in which the impact of missing those cookies is varied from cookie stealing, Cross-Site Scripting, to information disclosure.

Affected Hosts:

`https://non-prod.krungsriconsumer.com`

`https://uat.krungsriconsumer.com`

3.4.2 Steps to Reproduce

In this sample, when accessing a home page of the website, the HTTP returns with a set of HTTP headers which lack the required security headers. An attacker can observe the same response from the HTTP response which may result in further attacks.

- 1) URL: <https://non-prod.krungsriconsumer.com>

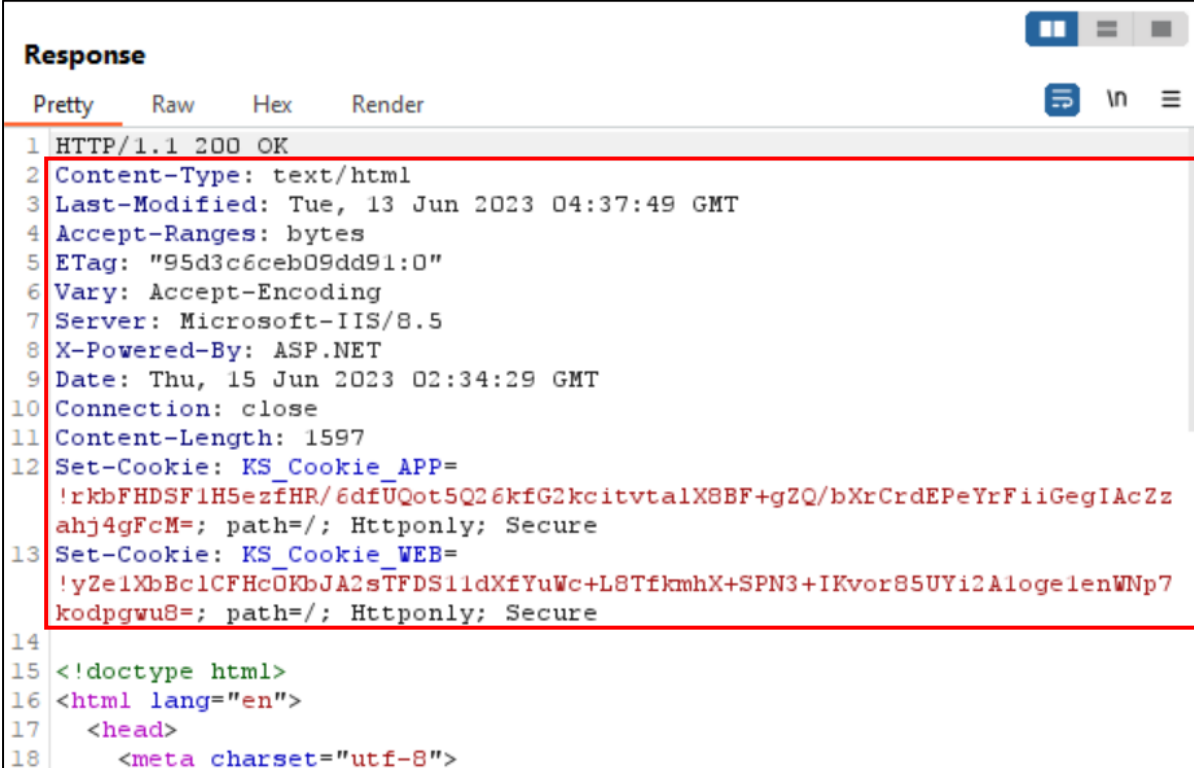


```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Thu, 15 Jun 2023 02:39:47 GMT
3 Content-Type: text/html; charset=utf-8
4 Connection: close
5 Server: Microsoft-IIS/10.0
6 Strict-Transport-Security: max-age=2592000
7 X-Powered-By: ASP.NET
8 Content-Length: 147430
9
10 <!DOCTYPE html>
11 <html lang="en">
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
```

Please note that multiple security headers are missing from the captured screenshot.

2) URL: <https://uat.krungsriconsumer.com>

The HTTP response shows more headers in this URL but still lacks the required security headers.



```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Content-Type: text/html
3 Last-Modified: Tue, 13 Jun 2023 04:37:49 GMT
4 Accept-Ranges: bytes
5 ETag: "95d3c6ceb09dd91:0"
6 Vary: Accept-Encoding
7 Server: Microsoft-IIS/8.5
8 X-Powered-By: ASP.NET
9 Date: Thu, 15 Jun 2023 02:34:29 GMT
10 Connection: close
11 Content-Length: 1597
12 Set-Cookie: KS_Cookie_APP=
!rkbFHDSF1H5ezfHR/6dfUQot5Q26kfG2kcitvtalX8BF+gZQ/bXrCrdEPeYrFiiGegIAcZz
ahj4gFcM=; path=/; Httponly; Secure
13 Set-Cookie: KS_Cookie_WEB=
!yZe1XbBclCFHcOKbJA2sTFDS11dXfYuWc+L8TfkmhX+SPN3+IKvor85UYi2AlogelenWNp7
kodpgwu8=; path=/; Httponly; Secure
14
15 <!doctype html>
16 <html lang="en">
17 <head>
18 <meta charset="utf-8">
```

3.4.3 Solution/Mitigation

The security headers listed below are required to be configured/added into the application.

'X-Content-Type-Options: nosniff'

This header instructs the browsers not to sniff data for MIME type but rely on Content-Type header.

'X-Frame-Options: SAMEORIGIN'

This header limits the application to only embedded into a frame from the same URL. But if the application is required to embedded into another website's frame, the below value can be used for the specific website. By the way, the ALLOW-FROM options are not supported in all browsers. In some unsupported browsers may use *X-Content-Security-Policy* instead.

ALL-FROM <https://www.yyy.com>

'X-XSS-Protection: 1; mode=block'

This header prevents the application of Cross-Site Scripting attacks. This header instructs the browser to stop pages from loading when reflected Cross-Site Scripting attack was detected.

Content-Security-Policy: default-src 'self' XXX ; object-src 'none'

For this header, make sure that all the required script are added within the XXX position, for example, if the application needs a script from Cloudflare to make the application running, add below value to the Content-Security-Policy header.

Default-src 'self' <https://cloudflare.com/path-to-your-script>; object-src 'none'

Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

This header instructs the browsers to only access the application using HTTPS. All sensitive data related application must set this header to ensure that all sensitive data will only be sent through the secured channel.

Cache-Control: no-cache, no-store, must-revalidate, max-age=0

Pragma: no-cache

Expires: 0

These headers come in a group. The above three lines are required to enable the function of the cache control. These headers limit the browsers to not store cache into the browser cache which help to prevent sensitive data disclosure or stored on the browser storage.

For more information, please see this link.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control>

3.4.4 Risk Calculation

Overall Risk: Low

Risk Formula	
Risk Level = Likelihood x Impact	

Risk Criteria and Level of Risk						
Risk Value		Likelihood Level				
		1	2	3	4	5
Impact Level	5	High (5)	High (10)	Critical (15)	Critical (20)	Critical (25)
	4	Medium (4)	High (8)	High (12)	Critical (16)	Critical (20)

	3	Low (3)	Medium (6)	High (9)	High (12)	Critical (15)
	2	Low (2)	Low (4)	Medium (6)	High (8)	High (10)
	1	Low (1)	Low (2)	Low (3)	Medium (4)	High (5)

CVSS Score
CVSS:3.1 N/A

4. Appendix

Port Scanning Result for non-prod.krungsriconsumer.com (10.68.144.6)

PORT	SERVICE	VERSION
443/tcp	ssl/https	Microsoft-Azure-Application-Gateway/v2

Port Scanning Result for uat.krungsriconsumer.com (192.168.43.154)

PORT	SERVICE	VERSION
443/tcp	ssl/http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)