



## Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.



High







Medium



Low

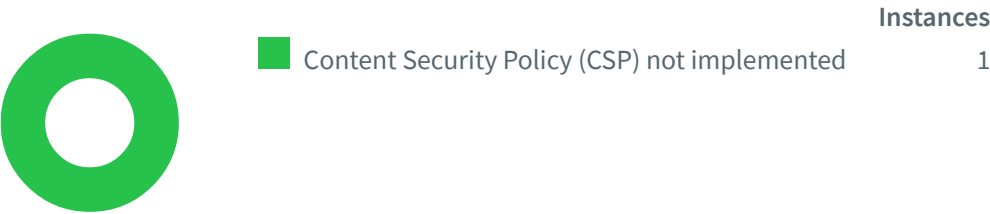


Informational

Severity	Vulnerabilities	Instances
 High	1	1
 Medium	1	1
 Low	2	2
 Informational	1	1
Total	5	5

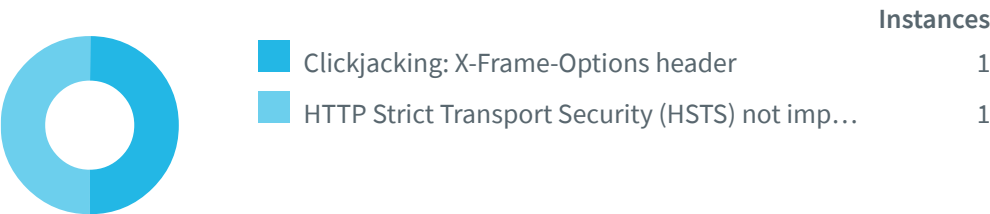
## Informational

---



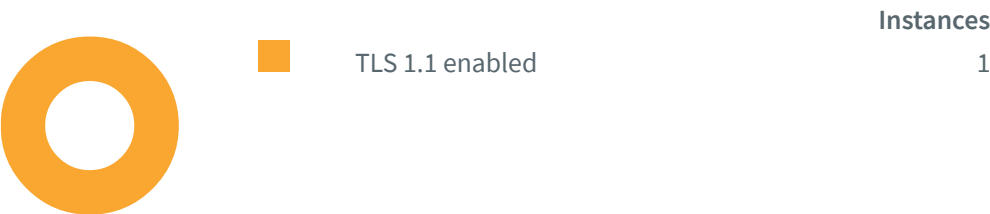
## Low Severity

---



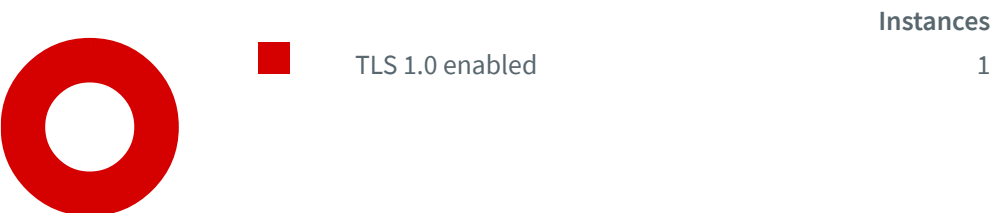
## Medium Severity

---








## High Severity

---



# Impacts

SEVERITY	IMPACT	
 High	<div>1</div>	TLS 1.0 enabled
 Medium	<div>1</div>	TLS 1.1 enabled
 Low	<div>1</div>	Clickjacking: X-Frame-Options header
 Low	<div>1</div>	HTTP Strict Transport Security (HSTS) not implemented
 Informational	<div>1</div>	Content Security Policy (CSP) not implemented

# TLS 1.0 enabled

---

The web server supports encryption through TLS 1.0, which was formally deprecated in March 2021 as a result of inherent security issues. In addition, TLS 1.0 is not considered to be "strong cryptography" as defined and required by the PCI Data Security Standard 3.2(.1) when used to protect sensitive information transferred to or from web sites. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.

## Impact

---

An attacker may be able to exploit this problem to conduct man-in-the-middle attacks and decrypt communications between the affected service and clients.

---

<https://svn.fsoft.com.vn/>

Confidence: 100%

The SSL server (port: 443) encrypts traffic using TLSv1.0.

## Recommendation

---

It is recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher.

## References

---

[RFC 8996: Deprecating TLS 1.0 and TLS 1.1](#)

<https://tools.ietf.org/html/rfc8996>

[Are You Ready for 30 June 2018? Saying Goodbye to SSL/early TLS](#)

<https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls>

[PCI 3.1 and TLS 1.2 \(Cloudflare Support\)](#)

<https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2>

# TLS 1.1 enabled

---

The web server supports encryption through TLS 1.1, which was formally deprecated in March 2021 as a result of inherent security issues. When aiming for Payment Card Industry (PCI) Data Security Standard (DSS) compliance, it is recommended to use TLS 1.2 or higher instead. According to PCI, "30 June 2018 is

the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.

## Impact

An attacker may be able to exploit this problem to conduct man-in-the-middle attacks and decrypt communications between the affected service and clients.

<https://svn.fsoft.com.vn/>

Confidence: 100%

The SSL server (port: 443) encrypts traffic using TLSv1.1.

## Recommendation

It is recommended to disable TLS 1.1 and replace it with TLS 1.2 or higher.

## References

[RFC 8996: Deprecating TLS 1.0 and TLS 1.1](https://tools.ietf.org/html/rfc8996)

<https://tools.ietf.org/html/rfc8996>

[Are You Ready for 30 June 2018? Saying Goodbye to SSL/early TLS](https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls)

<https://blog.pcisecuritystandards.org/are-you-ready-for-30-june-2018-sayin-goodbye-to-ssl-early-tls>

[PCI 3.1 and TLS 1.2 \(Cloudflare Support\)](https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2)

<https://support.cloudflare.com/hc/en-us/articles/205043158-PCI-3-1-and-TLS-1-2>

# Clickjacking: X-Frame-Options header

Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

The server did not return an **X-Frame-Options** header with the value DENY or SAMEORIGIN, which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into untrusted sites.

## Impact

---

The impact depends on the affected web application.

---

### <https://svn.fsoft.com.vn/>

Paths without secure XFO header:

- <https://svn.fsoft.com.vn/>
- <https://svn.fsoft.com.vn/svn/>
- <https://svn.fsoft.com.vn/index.html>

## Request

---

```
GET / HTTP/1.1
Referer: https://svn.fsoft.com.vn/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/99.0.4844.0 Safari/537.36
Host: svn.fsoft.com.vn
Connection: Keep-alive
```

## Recommendation

---

Configure your web server to include an X-Frame-Options header and a CSP header with frame-ancestors directive. Consult Web references for more information about the possible values for this header.

## References

---

### [The X-Frame-Options response header](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options)

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

### [Clickjacking](https://en.wikipedia.org/wiki/Clickjacking)

<https://en.wikipedia.org/wiki/Clickjacking>

### [OWASP Clickjacking](https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html)

[https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking\\_Defense\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html)

### [Frame Buster Buster](https://stackoverflow.com/questions/958997/frame-buster-buster-buster-code-needed)

<https://stackoverflow.com/questions/958997/frame-buster-buster-buster-code-needed>

# HTTP Strict Transport Security (HSTS) not implemented

---

HTTP Strict Transport Security (HSTS) tells a browser that a web site is only accessible using HTTPS. It was detected that your web application doesn't implement HTTP Strict Transport Security (HSTS) as the Strict Transport Security header is missing from the response.

## Impact

---

HSTS can be used to prevent and/or mitigate some types of man-in-the-middle (MitM) attacks

---

### <https://svn.fsoft.com.vn/>

URLs where HSTS is not enabled:

- <https://svn.fsoft.com.vn/svn/>

## Request

---

```
GET /svn/ HTTP/1.1
Referer: https://svn.fsoft.com.vn/
Cookie: __cf_bm=5QOJDbQWQR_jJ1_qLcrZFgtgMUyJRjWodbrUZf2s5Ws-1672893724-0-
AfYT6Q3hqjJsnTb4bQ+b942dZo/BnojrxWhg/YQP5IP4kdf62/r/Gk6uywY35Jqik+q3vdMLCqwy8jfpVUbmTWzTo0y/sLVEhATc
Awp05Ek1; _pk_id.319.9a00=8751f6527aa75043.1672893723.1.1672893723.1672893723.; _pk_ses.319.9a00=*
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/99.0.4844.0 Safari/537.36
Host: svn.fsoft.com.vn
Connection: Keep-alive
```

## Recommendation

---

It's recommended to implement HTTP Strict Transport Security (HSTS) into your web application. Consult web references for more information

## References

---

[hstspreload.org](https://hstspreload.org)

<https://hstspreload.org/>

[Strict-Transport-Security](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security)

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



# Content Security Policy (CSP) not implemented

---

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Content Security Policy (CSP) can be implemented by adding a **Content-Security-Policy** header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:

```
Content-Security-Policy:
default-src 'self';
script-src 'self' https://code.jquery.com;
```

It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

## Impact

---

CSP can be used to prevent and/or mitigate attacks that involve content/code injection, such as cross-site scripting/XSS attacks, attacks that require embedding a malicious resource, attacks that involve malicious use of iframes, such as clickjacking attacks, and others.

---

## <https://svn.fsoft.com.vn/>

Paths without CSP header:

- <https://svn.fsoft.com.vn/>
- <https://svn.fsoft.com.vn/index.html>
- <https://svn.fsoft.com.vn/svn/>

## Request

---

```
GET / HTTP/1.1
Referer: https://svn.fsoft.com.vn/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/99.0.4844.0 Safari/537.36
```

Host: svn.fsoft.com.vn

Connection: Keep-alive

## Recommendation

---

It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the **Content-Security-Policy** HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

## References

---

[Content Security Policy \(CSP\)](https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP)

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

[Implementing Content Security Policy](https://hacks.mozilla.org/2016/02/implementing-content-security-policy/)

<https://hacks.mozilla.org/2016/02/implementing-content-security-policy/>

## Coverage

---

<https://svn.fsoft.com.vn/>