



## Acunetix Threat Level 1

One or more low-severity type vulnerabilities have been discovered by the scanner.

### Scan Detail

Target	<a href="https://mp.ybx.greatcai.com">https://mp.ybx.greatcai.com</a>
Scan Type	Full Scan
Start Time	Apr 25, 2024, 5:23:38 PM GMT+8
Scan Duration	8 minutes
Requests	24932
Average Response Time	23ms
Maximum Response Time	21358ms



High







Medium



Low

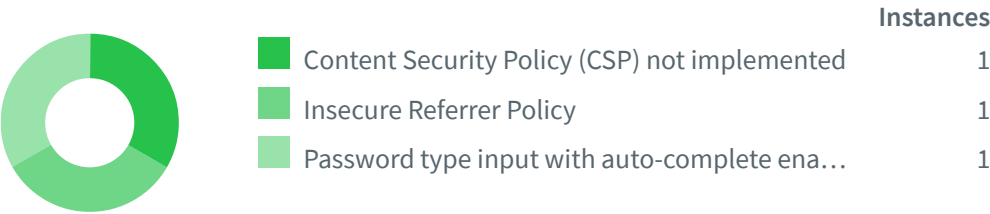


Informational

Severity	Vulnerabilities	Instances
 High	0	0
 Medium	0	0
 Low	6	6
 Informational	3	3
Total	9	9

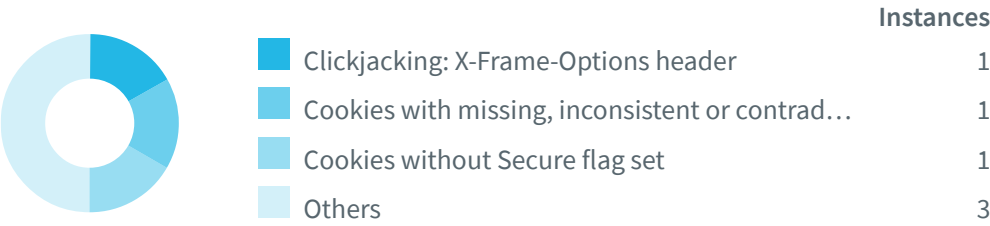
## Informational

---












## Low Severity

---



# Impacts

SEVERITY	IMPACT
 Low	<div>1</div> Clickjacking: X-Frame-Options header
 Low	<div>1</div> Cookies with missing, inconsistent or contradictory properties
 Low	<div>1</div> Cookies without Secure flag set
 Low	<div>1</div> HTTP Strict Transport Security (HSTS) not implemented
 Low	<div>1</div> Login page password-guessing attack
 Low	<div>1</div> TLS/SSL certificate about to expire
 Informational	<div>1</div> Content Security Policy (CSP) not implemented
 Informational	<div>1</div> Insecure Referrer Policy
 Informational	<div>1</div> Password type input with auto-complete enabled

# 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://mp.ybx.greatcai.com/>

Paths without secure XFO header:

- <https://mp.ybx.greatcai.com/Login/SendLoginSMSCode>
- <https://mp.ybx.greatcai.com/Login/LoginSMSCodeCheck>
- <https://mp.ybx.greatcai.com/Login>

## Request

---

```
POST /Login/SendLoginSMSCode HTTP/1.1
Host: mp.ybx.greatcai.com
Content-Length: 0
Pragma: no-cache
Cache-Control: no-cache
accept: application/json, text/javascript, */*; q=0.01
x-requested-with: XMLHttpRequest
accept-language: en-US
origin: https://mp.ybx.greatcai.com
cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8; ASP.NET_SessionId=flaionwtrgkjrpta2xsbbn11
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://mp.ybx.greatcai.com/Login
Accept-Encoding: gzip, deflate
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>

# Cookies with missing, inconsistent or contradictory properties

At least one of the following cookies properties causes the cookie to be invalid or incompatible with either a different property of the same cookie, of with the environment the cookie is being used in. Although this is not a vulnerability in itself, it will likely lead to unexpected behavior by the application, which in turn may cause secondary security issues.

## Impact

Cookies will not be stored, or submitted, by web browsers.

<https://mp.ybx.greatcai.com/>

Verified

List of cookies with missing, inconsistent or contradictory properties:

- <https://mp.ybx.greatcai.com/Login>

Cookie was set with:

```
Set-Cookie: ASP.NET_SessionId=flaionwtrgkjrpta2xsbbn11; path=/; HttpOnly
```

This cookie has the following issues:

- Cookie without SameSite attribute.

When cookies lack the SameSite attribute, Web browsers may apply different and sometimes unexpected defaults. It is therefore recommended to add a SameSite attribute with an appropriate value of either "Strict", "Lax", or "None".

## Request

---

```
GET /Login HTTP/1.1
Host: mp.ybx.greatcai.com
Pragma: no-cache
Cache-Control: no-cache
upgrade-insecure-requests: 1
accept-language: en-US
accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip,deflate
Cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4298.0 Safari/537.36
```

## Recommendation

---

Ensure that the cookies configuration complies with the applicable standards.

## References

---

[MDN | Set-Cookie](#)

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

[Securing cookies with cookie prefixes](#)

<https://www.sjoerdlangkemper.nl/2017/02/09/cookie-prefixes/>

[Cookies: HTTP State Management Mechanism](#)

<https://tools.ietf.org/html/draft-ietf-httpbis-rfc6265bis-05>

[SameSite Updates - The Chromium Projects](#)

<https://www.chromium.org/updates/same-site>

[draft-west-first-party-cookies-07: Same-site Cookies](#)

<https://tools.ietf.org/html/draft-west-first-party-cookies-07>

# Cookies without Secure flag set

---

One or more cookies does not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL/TLS channels. This is an important security protection for session cookies.

## Impact

---

Cookies could be sent over unencrypted channels.

---

<https://mp.ybx.greatcai.com/>

Verified

Cookies without Secure flag set:

- <https://mp.ybx.greatcai.com/Login>

```
Set-Cookie: ASP.NET_SessionId=flaionwtrgkjrpta2xsbbn11; path=/; HttpOnly
```

## Request

---

```
GET /Login HTTP/1.1
Host: mp.ybx.greatcai.com
Pragma: no-cache
Cache-Control: no-cache
upgrade-insecure-requests: 1
accept-language: en-US
accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4298.0 Safari/537.36
```

## Recommendation

---

If possible, you should set the Secure flag for these cookies.

# 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://mp.ybx.greatcai.com/>

URLs where HSTS is not enabled:

- <https://mp.ybx.greatcai.com/Login/SendLoginSMSCode>
- <https://mp.ybx.greatcai.com/Login/LoginSMSCodeCheck>
- <https://mp.ybx.greatcai.com/Login>

## Request

---

```
POST /Login/SendLoginSMSCode HTTP/1.1
Host: mp.ybx.greatcai.com
Content-Length: 0
Pragma: no-cache
Cache-Control: no-cache
accept: application/json, text/javascript, */*; q=0.01
x-requested-with: XMLHttpRequest
accept-language: en-US
origin: https://mp.ybx.greatcai.com
cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8; ASP.NET_SessionId=f1aionwtrgkjrpta2xsbbn11
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://mp.ybx.greatcai.com/Login
Accept-Encoding: gzip, deflate
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4298.0 Safari/537.36
```



## 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>

# Login page password-guessing attack

---

A common threat web developers face is a password-guessing attack known as a brute force attack. A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until you discover the one correct combination that works.

This login page doesn't have any protection against password-guessing attacks (brute force attacks). It's recommended to implement some type of account lockout after a defined number of incorrect password attempts. Consult Web references for more information about fixing this problem.

## Impact

---

An attacker may attempt to discover a weak password by systematically trying every possible combination of letters, numbers, and symbols until it discovers the one correct combination that works.

<https://mp.ybx.greatcai.com/Login>

Confidence: 80%

## Request

---

```
POST /Login HTTP/1.1
Referer: https://mp.ybx.greatcai.com/Login
Cookie: __jsluid_s=3d62f4212fe330a00d5dc0148d4eee7c;
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 43
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4298.0 Safari/537.36
Host: mp.ybx.greatcai.com
Connection: Keep-alive
```

## Recommendation

---

It's recommended to implement some type of account lockout after a defined number of incorrect password attempts.

## References

---

### [Blocking Brute Force Attacks](#)

[https://www.owasp.org/index.php/Blocking\\_Brute\\_Force\\_Attacks](https://www.owasp.org/index.php/Blocking_Brute_Force_Attacks)

# TLS/SSL certificate about to expire

---

One of the TLS/SSL certificates used by your server is about to expire.

Once the certificate has expired, most web browsers will present end-users with a security warning, asking them to manually confirm the authenticity of your certificate chain. Software or automated systems may silently refuse to connect to the server.

This alert is not necessarily caused by the server (leaf) certificate, but may have been triggered by an intermediate certificate. Please refer to the certificate serial number in the alert details to identify the affected certificate.

## Impact

---

If an application server detects an expired certificate with a system it is communicating with, the application server may continue processing data as if nothing happened, or the connection may be abruptly terminated.

---

<https://mp.ybx.greatcai.com/>

Confidence: 100%

Error: could not render details.

## Recommendation

---

Contact your Certificate Authority to renew the SSL certificate.

# 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://mp.ybx.greatcai.com/>

Paths without CSP header:

- <https://mp.ybx.greatcai.com/Login>

## Request

---

```
GET /Login HTTP/1.1
Host: mp.ybx.greatcai.com
Pragma: no-cache
Cache-Control: no-cache
upgrade-insecure-requests: 1
accept-language: en-US
accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
```

```
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4298.0 Safari/537.36
```

## 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>

[Implementing Content Security Policy](#)

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

# Insecure Referrer Policy

---

Referrer Policy controls behaviour of the Referer header, which indicates the origin or web page URL the request was made from. The web application uses insecure Referrer Policy configuration that may leak user's information to third-party sites.

## Impact

---

In some situations, an attacker may leak a user's private data

**<https://mp.ybx.greatcai.com/>**

URLs where Referrer Policy configuration is insecure:

- <https://mp.ybx.greatcai.com/Login/SendLoginSMSCode>
- <https://mp.ybx.greatcai.com/Login/LoginSMSCodeCheck>
- <https://mp.ybx.greatcai.com/Login>

## Request

---

```
POST /Login/SendLoginSMSCode HTTP/1.1
Host: mp.ybx.greatcai.com
Content-Length: 0
Pragma: no-cache
Cache-Control: no-cache
accept: application/json, text/javascript, */*; q=0.01
x-requested-with: XMLHttpRequest
accept-language: en-US
origin: https://mp.ybx.greatcai.com
cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8; ASP.NET_SessionId=flaionwtrgkjrpta2xsbbn11
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://mp.ybx.greatcai.com/Login
Accept-Encoding: gzip, deflate
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4298.0 Safari/537.36
```

## Recommendation

---

Consider setting Referrer-Policy header to 'strict-origin-when-cross-origin' or a stricter value

## References

---

### [Referrer-Policy](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy)

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

# Password type input with auto-complete enabled

---

When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved. Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.

## Impact

---

Possible sensitive information disclosure.

### <https://mp.ybx.greatcai.com/>

Pages with auto-complete password inputs:

- <https://mp.ybx.greatcai.com/Login>

```
Form name: <empty>
Form action: /Login
Form method: POST
Password input: password
```

## Request

---

```
POST /Login HTTP/1.1
Host: mp.ybx.greatcai.com
Content-Length: 63
Pragma: no-cache
Cache-Control: no-cache
upgrade-insecure-requests: 1
origin: https://mp.ybx.greatcai.com
content-type: application/x-www-form-urlencoded
accept-language: en-US
accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
cookie: __jsluid_s=6eb7fd68e778306c57a4196a7349a6e8; ASP.NET_SessionId=flaionwtrgkjrpta2xsbbn11
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://mp.ybx.greatcai.com/Login
Accept-Encoding: gzip,deflate
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/88.0.4298.0 Safari/537.36

userName=songjj%40YBX&password=E8F8A55DB55D265B7963468366037E06
```

## Recommendation

---

The password auto-complete should be disabled in sensitive applications.

To disable auto-complete, you may use a code similar to:

```
<INPUT TYPE="password" AUTOCOMPLETE="off">
```

# Coverage

 https://mp.ybx.greatcai.com

 admin

 api

 axis2

 axis2-admin

 welcome

 cacti

 cognos\_express

 manager

 html

 console

 Content

 bootstrap.min.css

 login.css

 Site.css

 extrahop

 fonts

 host-manager

 html

 text

 images

 lc

 system

 console

 Login

 Index

 LoginSMSCheck

 Inputs

 code,

 Main

 SendLoginSMSCode

- manager
  - html
  - status
- nagios
- opennms
  - login.jsp
- otrs
- rockmongo
- Scripts
  - bootstrap.min.js
  - jquery-3.7.1.min.js
  - jquery.md5.js
- server
  - TCPIPGEN.htm
- system
  - console
- tomcat
  - host-manager
    - html
    - text
  - manager
    - html
    - status
- ui
  - authentication
- webtools
- zabbix
- applet.html
- index.asp
- index.html
- Login
  - Inputs
    - POST password, userName
- login.html



---

 login.jsp

---

 login.php