



Group IT\_App-Sec Report 25 Apr 2025

Each targeted web application is listed with the total number of detected vulnerabilities and sensitive content.

Neveille Mehta

Capgemini  
Capgemini knowledge Park(SEZ), T3/IT4, Airoli knowledge Park, Thane  
Belapur Rd, Airoli.  
Navi Mumbai, Maharashtra 400708  
India

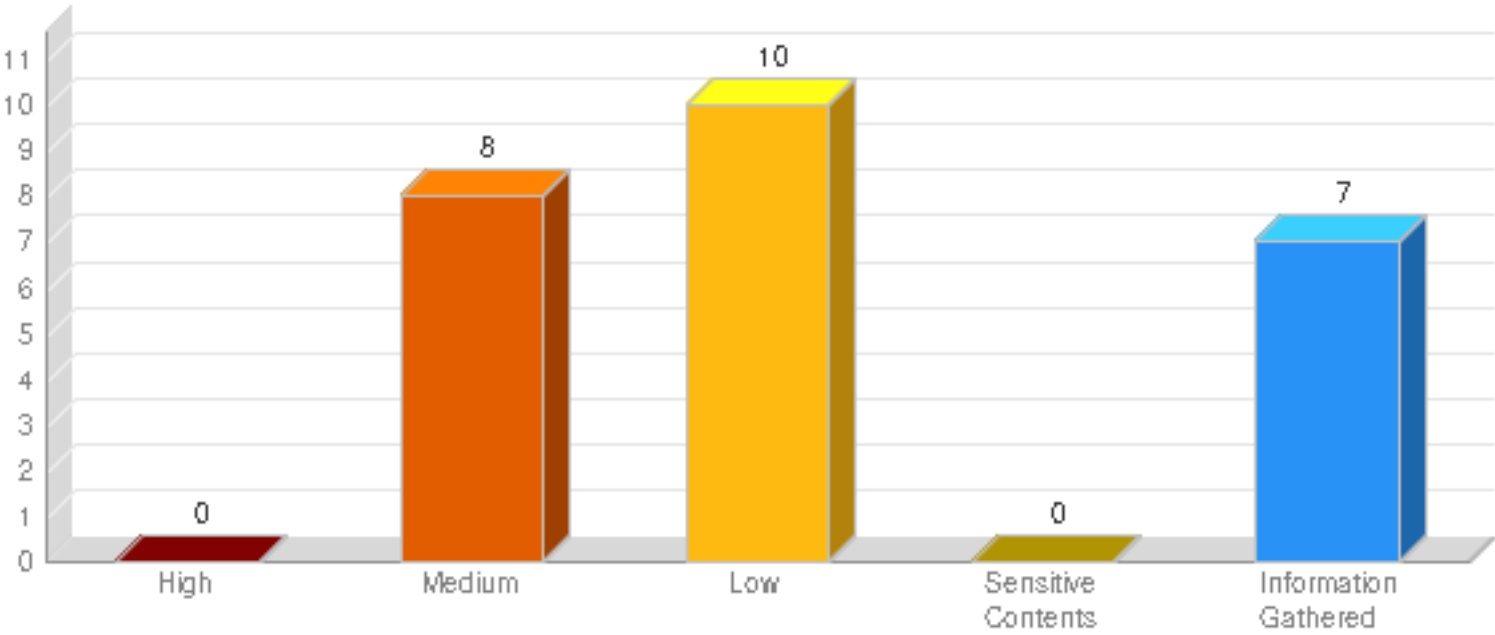
Target and Filters

Web Applications (1)	Group IT_Prism_UAT
Inclusion search lists	Include_Confirmed_WAS Vulnerabilities, Include_Informational_WAS Vulnerabilities
Status	New, Active, Re-Opened
Detection Source	Qualys, Burp

Summary

Security Risk	Web Applications	Vulnerabilities	Sensitive Contents	Information Gathered
MED	1	18	0	7

Findings by Severity

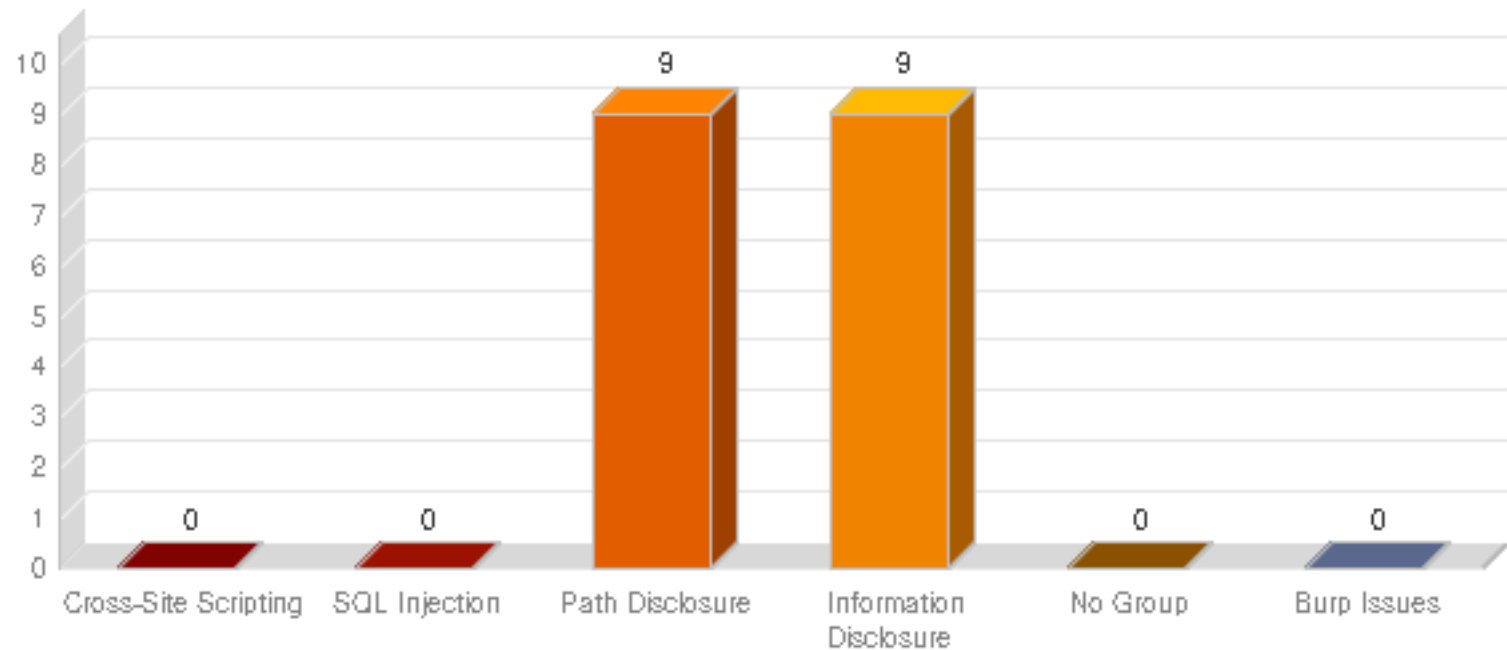


Vulnerabilities by Status

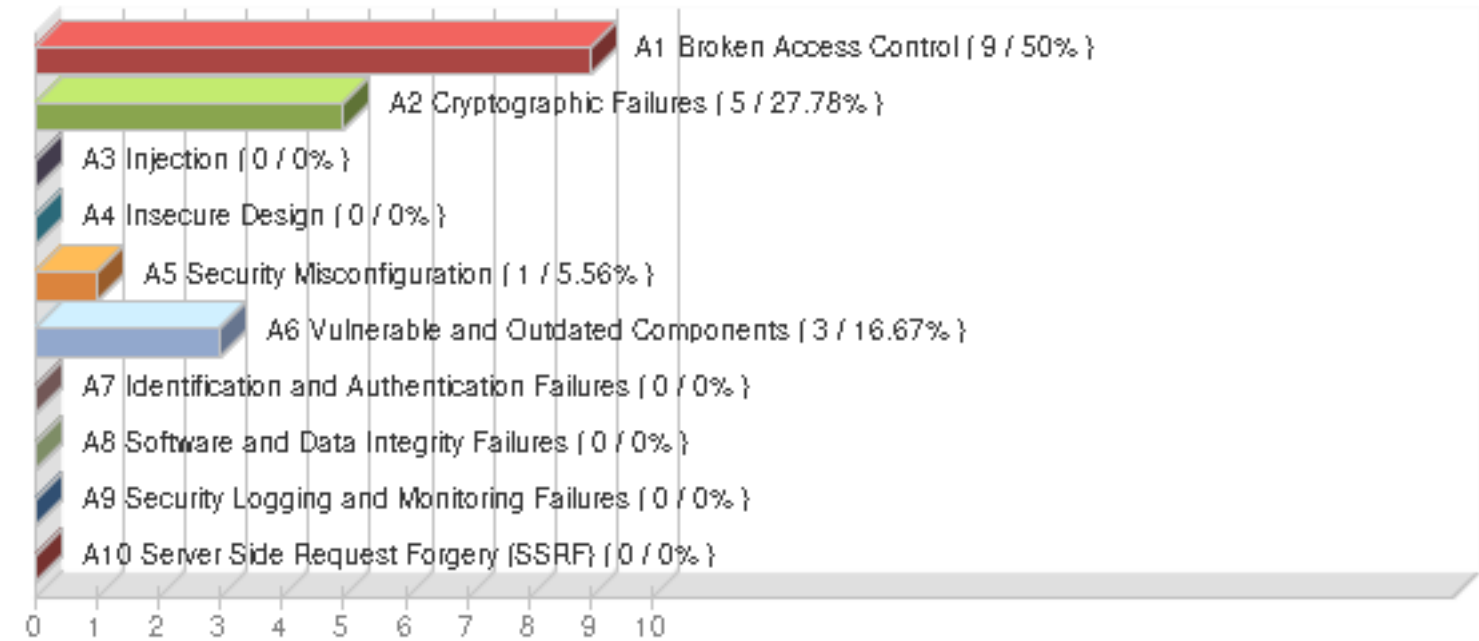


# WAS Web Application Report

Vulnerabilities by Group



OWASP Top 10 2021 Vulnerabilities



Web Application	High	Medium	Low	Sensitive Contents	Information Gathered
Group IT_Prism_UAT	0	8	10	0	7

Results(25)

Vulnerability (18)

Path Disclosure (9)

LOW 150004 Predictable Resource Location Via Forced Browsing (1)

LOW 150004 Predictable Resource Location Via Forced Browsing

Group IT\_Prism\_UATActive

URL: https://prism-preprod.capgemini.com/phpMyAdmin/ChangeLog

Finding #	33404804	Severity	Confirmed Vulnerability - Level 2
Unique #	b3b876c7-75a0-4db1-92d9-fc760612fd26		
Group	Path Disclosure	First Time Detected	09 Sep 2024 14:40 GMT+0630
CWE	CWE-22	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A1 Broken Access Control	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	WASC-15 APPLICATION MISCONFIGURATION WASC-16 DIRECTORY INDEXING WASC-17 IMPROPER FILESYSTEM PERMISSIONS	Times Detected	4
CVSS V3 Base	5.3	CVSS V3 Temporal	4.7
CVSS V3 Attack Vector Network			

Details

Threat

A file, directory, or directory listing was discovered on the Web server. These resources are confirmed to be present based on our logic. Some of the content on these files might have sensitive information.

NOTE: Links found in 150004 are found by forced crawling so will not automatically be added to 150009 Links Crawled or the application site map. If links found in 150004 need to be tested they must be added as Explicit URI so they are included in scope and then will be reported in 150009. Once the link is added to be in scope (i.e. Explicit URI) this same link will no longer be reported for 150004.

Impact

The contents of this file or directory may disclose sensitive information.

Solution

It is advised to review the contents of the disclosed files. If the contents contain sensitive information, please verify that access to this file or directory is permitted. If necessary, remove it or apply access controls to it.

Detection Information

Parameter	No param has been required for detecting the information.
Authentication	In order to detect this vulnerability, the scan required authentication to be enabled.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/

Payloads (1 instance)

#1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/ChangeLog  
Referer: https://prism-preprod.capgemini.com/  
Cookie: pma\_lang\_https=en; phpMyAdmin\_https=cvn1ruh1e0sfriu5dgvhhd6ijb; PFSTG=PmqBvRSwbIYs4NVAPFamAoirO4rG7u2tPJSV8v1765kF;  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36  
Host: prism-preprod.capgemini.com  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

comment:  
Original URL is: https://prism-preprod.capgemini.com/phpMyAdmin/  
  
HTTP/1.1 200 OK

LOW 150023 Directory Listing (4)  
LOW 150023 Directory Listing

Group IT\_Prism\_UAT New

URL: https://prism-preprod.capgemini.com/icons/			
Finding #	36277046	Severity	Confirmed Vulnerability - Level 2
Unique #	4e409203-2237-4829-a4a6-65c736aad55		
Group	Path Disclosure	First Time Detected	24 Apr 2025 14:53 GMT+0630
CWE	CWE-548	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A1 Broken Access Control	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	WASC-16 DIRECTORY INDEXING	Times Detected	1
CVSS V3 Base	5.3	CVSS V3 Temporal5	CVSS V3 Attack Vector Network

Details

**Threat**  
The Web server presents a directory listing.

**Impact**  
All file names in this directory are exposed.

**Solution**  
The presence of a browseable directory does not necessarily imply a vulnerability. Determine if the directory listing is intended to be displayed. Verify that no files in the directory contain content that should not be served by the Web application.

Detection Information

Parameter	No param has been required for detecting the information.
Authentication	In order to detect this vulnerability, the scan required authentication to be enabled.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/

Payloads (1 instance)

#1 Request

GET https://prism-preprod.capgemini.com/icons/  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

comment: This directory was discovered during the crawl phase.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html><head>
<title>Index of /icons</title>
</head>
<body>
<h1>Index of /icons</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><
```

LOW

150023 Directory Listing

Group IT\_Prism\_UAT

New

URL: https://prism-preprod.capgemini.com/icons/?C=S%3BOA

Finding #	36276236	Severity	Confirmed Vulnerability - Level 2
Unique #	30aae04b-d6af-4172-83d6-8e44972765d7		
Group	Path Disclosure	First Time Detected	24 Apr 2025 14:53 GMT+0630
CWE	CWE-548	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A1 Broken Access Control	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	WASC-16 DIRECTORY INDEXING	Times Detected	1
CVSS V3 Base	5.3	CVSS V3 Temporal5	CVSS V3 Attack Vector Network

Details

Threat

The Web server presents a directory listing.

Impact

All file names in this directory are exposed.

Solution

The presence of a browseable directory does not necessarily imply a vulnerability. Determine if the directory listing is intended to be displayed. Verify that no files in the directory contain content that should not be served by the Web application.

Detection Information

Parameter	No param has been required for detecting the information.
Authentication	In order to detect this vulnerability, the scan required authentication to be enabled.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/  
https://prism-preprod.capgemini.com/icons/

Payloads (4 instances)

#1 Request

GET https://prism-preprod.capgemini.com/icons/?C=S%3BOA  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*  
  
*Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.*

#1 Response

comment: This directory was discovered during the crawl phase.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html><head>
<title>Index of /icons</title>
</head>
<body>
<h1>Index of /icons</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><
```

#2 Request

GET https://prism-preprod.capgemini.com/icons/?C=M%3BOA  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*  
  
*Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.*

#2 Response

comment: This directory was discovered during the crawl phase.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html><head>
<title>Index of /icons</title>
</head>
<body>
<h1>Index of /icons</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><
```

#3 Request

GET https://prism-preprod.capgemini.com/icons/?C=N%3BOD  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlIxIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#3 Response

comment: This directory was discovered during the crawl phase.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html><head>
<title>Index of /icons</title>
</head>
<body>
<h1>Index of /icons</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><
```

#4 Request

GET https://prism-preprod.capgemini.com/icons/?C=D%3BOA  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlIxIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#4 Response

comment: This directory was discovered during the crawl phase.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html><head>
<title>Index of /icons</title>
</head>
<body>
<h1>Index of /icons</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><
```

LOW150023 Directory Listing

GroupIT\_Prism\_UATActive

URL: https://prism-preprod.capgemini.com/icons/small/			
Finding #	33404802	Severity	Confirmed Vulnerability - Level 2
Unique #	e311557e-76ea-4439-aa31-ea9b9d980c1f		
Group	Path Disclosure	First Time Detected	09 Sep 2024 14:40 GMT+0630
CWE	CWE-548	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A1 Broken Access Control	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	WASC-16 DIRECTORY INDEXING	Times Detected	3
CVSS V3 Base	5.3	CVSS V3 Temporal	5
		CVSS V3 Attack Vector	Network



Details

Threat

The Web server presents a directory listing.

Impact

All file names in this directory are exposed.

Solution

The presence of a browseable directory does not necessarily imply a vulnerability. Determine if the directory listing is intended to be displayed. Verify that no files in the directory contain content that should not be served by the Web application.

Detection Information

Parameter	No param has been required for detecting the information.
Authentication	In order to detect this vulnerability, the scan required authentication to be enabled.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/

Payloads (2 instances)

#1 Request

GET https://prism-preprod.capgemini.com/icons/small/  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

comment: This directory was discovered during the crawl phase.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><html><head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tbody><tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top">

# WAS Web Application Report

- Set response header X-Frame-Options: deny
- Set response header X-Content-Type-Options: nosniff.

## Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |
| Access Path    | Here is the path followed by the scanner to reach the exploitable URL:                 |

https://prism-preprod.capgemini.com/

## Payloads (2 instances)

### #1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/  
Referer: https://prism-preprod.capgemini.com/  
Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

### #1 Response

Relative Path CSS Links found:  
<link rel="stylesheet" type="text/css" href="/themes/pmahomme/css/theme.css?v=5.2.0">  
<link rel="stylesheet" type="text/css" href="/themes/pmahomme/jquery/jquery-ui.css">  
<link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/addon/hint/show-hint.css?v=5.2.0">  
<link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/addon/lint/lint.css?v=5.2.0">  
<link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/lib/codemirror.css?v=5.2.0">

### #2 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/  
Referer: https://prism-preprod.capgemini.com/  
Cookie: pma\_lang\_https=en; phpMyAdmin\_https=0o0lmgue2nnq43nnceka18p230; PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

### #2 Response

Relative Path CSS Links found:  
<link rel="stylesheet" type="text/css" href="/themes/pmahomme/css/theme.css?v=5.2.0">  
<link rel="stylesheet" type="text/css" href="/themes/pmahomme/jquery/jquery-ui.css">  
<link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/addon/hint/show-hint.css?v=5.2.0">  
<link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/addon/lint/lint.css?v=5.2.0">  
<link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/lib/codemirror.css?v=5.2.0">  
Please check there may be more pages with relative path CSS links.

**LOW** 150246 Path-relative stylesheet import (PRSSI) vulnerability

Group IT\_Prism\_UAT **New**

URL: https://prism-preprod.capgemini.com/phpMyAdmin/doc/html/index.html

|           |          |          |                                   |
|-----------|----------|----------|-----------------------------------|
| Finding # | 36276234 | Severity | Confirmed Vulnerability - Level 1 |
|-----------|----------|----------|-----------------------------------|

# WAS Web Application Report

|                       |                                      |                     |                            |
|-----------------------|--------------------------------------|---------------------|----------------------------|
| Unique #              | 51e38d37-14c0-48be-8c59-ce758aa4960f |                     |                            |
| Group                 | Path Disclosure                      | First Time Detected | 24 Apr 2025 14:53 GMT+0630 |
| CWE                   | CWE-23                               | Last Time Detected  | 24 Apr 2025 14:53 GMT+0630 |
| OWASP                 | A1 Broken Access Control             | Last Time Tested    | 24 Apr 2025 14:53 GMT+0630 |
| WASC                  | -                                    | Times Detected      | 1                          |
| CVSS V3 Base          | 3.1                                  | CVSS V3 Temporal    | 2.9                        |
| CVSS V3 Attack Vector |                                      | Network             |                            |

## Details

### Threat

Relative URLs can be dangerous since browser may not determine the correct directory. If the HTML uses path-relative CSS links, it may be susceptible to path-relative stylesheet import (PRSSI) vulnerabilities. This could allow an attacker to take advantage of CSS imports with relative URLs by overwriting their target file.

References:  
[Evil CSS Injection](#)  
[Relative Path Overwrite Attack](#)  
[Research paper: Large-Scale Analysis of Style Injection by Relative Path Overwrite](#)

### Impact

An attacker may trick browsers into importing JavaScript or HTML code as a stylesheet. This has been shown to enable a number of different attacks, including cross-site scripting (XSS) and exfiltration of CSRF tokens.

### Solution

It is recommended to use absolute URLs for CSS imports. Alternately you can add the HTML "base" tag in the document which defines the base URL or target location for all the relative URLs.

The vulnerability can also be mitigated by using the following best practices to harden the web pages:

- Set a DOCTYPE which does not allow Quirks mode as explained at <https://hsivonen.fi/doctype/>
- Set response header X-Frame-Options: deny
- Set response header X-Content-Type-Options: nosniff.

## Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

## Payloads (1 instance)

### #1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/doc/html/index.html  
Referer: https://prism-preprod.capgemini.com/  
Cookie: pma\_lang\_https=en; phpMyAdmin\_https=cv7ottnismdintleegu6tmdk9; PFSTG=JuRNN5Uihfbt1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

### #1 Response

Relative Path CSS Links found:  
<link rel="stylesheet" href="\_static/classic.css" type="text/css">  
<link rel="stylesheet" href="\_static/pygments.css" type="text/css">

LOW

150246 Path-relative stylesheet import (PRSSI) vulnerability

Group IT\_Prism\_UAT

New

URL: https://prism-preprod.capgemini.com/phpMyAdmin/doc/html/search.html?q=1

|              |                                      |                       |                                   |
|--------------|--------------------------------------|-----------------------|-----------------------------------|
| Finding #    | 36277056                             | Severity              | Confirmed Vulnerability - Level 1 |
| Unique #     | f3861de1-ca54-4d75-933c-30cc13d47352 |                       |                                   |
| Group        | Path Disclosure                      | First Time Detected   | 24 Apr 2025 14:53 GMT+0630        |
| CWE          | CWE-23                               | Last Time Detected    | 24 Apr 2025 14:53 GMT+0630        |
| OWASP        | A1 Broken Access Control             | Last Time Tested      | 24 Apr 2025 14:53 GMT+0630        |
| WASC         | -                                    | Times Detected        | 1                                 |
| CVSS V3 Base | 3.1                                  | CVSS V3 Temporal      | 2.9                               |
|              |                                      | CVSS V3 Attack Vector | Network                           |

Details

Threat

An relative URLs can be dangerous since browser may not determine the correct directory. If the HTML uses path-relative CSS links, it may be susceptible to path-relative stylesheet import (PRSSI) vulnerabilities. This could allow an attacker to take advantage of CSS imports with relative URLs by overwriting their target file.

References:

[Evil CSS Injection](#)  
[Relative Path Overwrite Attack](#)  
[Research paper: Large-Scale Analysis of Style Injection by Relative Path Overwrite](#)

Impact

An attacker may trick browsers into importing JavaScript or HTML code as a stylesheet. This has been shown to enable a number of different attacks, including cross-site scripting (XSS) and exfiltration of CSRF tokens.

Solution

It is recommended to use absolute URLs for CSS imports. Alternately you can add the HTML "base" tag in the document which defines the base URL or target location for all the relative URLs.

The vulnerability can also be mitigated by using the following best practices to harden the web pages:

- Set a DOCTYPE which does not allow Quirks mode as explained at https://hsivonen.fi/doctype/
- Set response header X-Frame-Options: deny
- Set response header X-Content-Type-Options: nosniff.

Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

Payloads (1 instance)



#1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/doc/html/search.html?q=1  
Referer: https://prism-preprod.capgemini.com/  
Cookie: pma\_lang\_https=en; phpMyAdmin\_https=cv7ottvnismdintleegu6tmdk9; PFSTG=JuRNN5Uihfbt1aKei30zN0c68PD0CAqzXlxbIenWzUmp;  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*  
Content-Length: 3

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Relative Path CSS Links found:  
<link rel="stylesheet" href="\_static/classic.css" type="text/css">  
<link rel="stylesheet" href="\_static/pygments.css" type="text/css">

LOW

150246 Path-relative stylesheet import (PRSSI) vulnerability

Group IT\_Prism\_UAT

New

URL: https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/%26route=%2F%26token=21503a5f66643f70526c2e645b2e2956%26lang=en

|              |                                      |                     |                                   |
|--------------|--------------------------------------|---------------------|-----------------------------------|
| Finding #    | 36277040                             | Severity            | Confirmed Vulnerability - Level 1 |
| Unique #     | 39982631-4921-448a-a309-aef4662c008e |                     |                                   |
| Group        | Path Disclosure                      | First Time Detected | 24 Apr 2025 14:53 GMT+0630        |
| CWE          | CWE-23                               | Last Time Detected  | 24 Apr 2025 14:53 GMT+0630        |
| OWASP        | A1 Broken Access Control             | Last Time Tested    | 24 Apr 2025 14:53 GMT+0630        |
| WASC         | -                                    | Times Detected      | 1                                 |
| CVSS V3 Base | 3.1                                  | CVSS V3 Temporal2.9 | CVSS V3 Attack VectorNetwork      |

Details

Threat

Relative URLs can be dangerous since browser may not determine the correct directory. If the HTML uses path-relative CSS links, it may be susceptible to path-relative stylesheet import (PRSSI) vulnerabilities. This could allow an attacker to take advantage of CSS imports with relative URLs by overwriting their target file.

References:  
[Evil CSS Injection](#)  
[Relative Path Overwrite Attack](#)  
[Research paper: Large-Scale Analysis of Style Injection by Relative Path Overwrite](#)

Impact

An attacker may trick browsers into importing JavaScript or HTML code as a stylesheet. This has been shown to enable a number of different attacks, including cross-site scripting (XSS) and exfiltration of CSRF tokens.

Solution

It is recommended to use absolute URLs for CSS imports. Alternately you can add the HTML "base" tag in the document which defines the base URL or target location for all the relative URLs.

The vulnerability can also be mitigated by using the following best practices to harden the web pages:

- Set a DOCTYPE which does not allow Quirks mode as explained at <https://hsivonen.fi/doctype/>
- Set response header X-Frame-Options: deny
- Set response header X-Content-Type-Options: nosniff.

Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

Payloads (2 instances)

#1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&a mp;token=21503a5f66643f70526c2e645b2e2956&lang=en

Referer: https://prism-preprod.capgemini.com/

Cookie: pma\_lang\_https=en; phpMyAdmin\_https=0pgntgl623qgotjoj3ak6cnjt4; PFSTG=JuRNN5Uihfbt1aKei30zN0c68PD0CAqzXlxbIenWzUmp;

Host: prism-preprod.capgemini.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15

Accept: \*/\*

Content-Length: 56

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Relative Path CSS Links found:

<link rel="stylesheet" type="text/css" href="/themes/pmahomme/css/theme.css?v=5.2.0">

<link rel="stylesheet" type="text/css" href="/themes/pmahomme/jquery/jquery-ui.css">

<link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/hint/show-hint.css?v=5.2.0">

<link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/lint/lint.css?v=5.2.0">

<link rel="stylesheet" type="text/css" href="js/vendor/codemirror/lib/codemirror.css?v=5.2.0">

#2 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en& :route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329

Referer: https://prism-preprod.capgemini.com/

Cookie: pma\_lang\_https=en; phpMyAdmin\_https=q3k1033338saekjrk4akfle6q4; PFSTG=JuRNN5Uihfbt1aKei30zN0c68PD0CAqzXlxbIenWzUmp;

Host: prism-preprod.capgemini.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15

Accept: \*/\*

Content-Length: 56

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#2 Response

Relative Path CSS Links found:

<link rel="stylesheet" type="text/css" href="/themes/pmahomme/css/theme.css?v=5.2.0">

<link rel="stylesheet" type="text/css" href="/themes/pmahomme/jquery/jquery-ui.css">

<link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/hint/show-hint.css?v=5.2.0">

<link rel="stylesheet" type="text/css" href="js/vendor/codemirror/addon/lint/lint.css?v=5.2.0">

<link rel="stylesheet" type="text/css" href="js/vendor/codemirror/lib/codemirror.css?v=5.2.0">

Information Disclosure (9)

MED

150150 HTML form containing password field(s) is served over HTTP (4)

MED

150150 HTML form containing password field(s) is served over HTTP

Group IT\_Prism\_UAT

New

URL: http://prism-preprod.capgemini.com/phpMyAdmin/

|           |                                                |                     |                                   |
|-----------|------------------------------------------------|---------------------|-----------------------------------|
| Finding # | 36276232                                       | Severity            | Confirmed Vulnerability - Level 3 |
| Unique #  | 2d62aba6-ee1b-493f-8a6d-6825f6a4c1c3           |                     |                                   |
| Group     | Information Disclosure                         | First Time Detected | 24 Apr 2025 14:53 GMT+0630        |
| CWE       | CWE-523                                        | Last Time Detected  | 24 Apr 2025 14:53 GMT+0630        |
| OWASP     | A2 Cryptographic Failures                      | Last Time Tested    | 24 Apr 2025 14:53 GMT+0630        |
| WASC      | WASC-4 INSUFFICIENT TRANSPORT LAYER PROTECTION | Times Detected      | 1                                 |

Details

Threat

An HTML form containing a password field (input field with type=password) is served insecurely over HTTP.

Impact

Serving an HTML form containing a password field (type=password) makes it susceptible to phishing attack. Failure to utilize SSL/TLS for the login landing page could allow an attacker to modify the login form action, causing the user's credentials to be posted to an arbitrary location.

Solution

Make sure that HTML forms containing password field(s) are always served over HTTPS only.

References:

- CWE-523 (Unprotected Transport of Credentials): <https://cwe.mitre.org/data/definitions/523.html>
- OWASP Transport Layer Protection Cheat Sheet (see "Use TLS for All Pages"): [https://cheatsheetseries.owasp.org/cheatsheets/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html)

Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

Payloads (1 instance)

#1 Request

GET http://prism-preprod.capgemini.com/phpMyAdmin/  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Form with credentials is served over insecure connection.  
Form action URI: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=  
  
Password fields on the form: pma\_password  
served on insecure (i.e. HTTP) page: http://prism-preprod.capgemini.com/phpMyAdmin/

MED

150150 HTML form containing password field(s) is served over HTTP

Group IT\_Prism\_UAT

New

URL: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/

|              |                                                |                       |                                   |
|--------------|------------------------------------------------|-----------------------|-----------------------------------|
| Finding #    | 36277044                                       | Severity              | Confirmed Vulnerability - Level 3 |
| Unique #     | e9e6976a-c308-4b5a-bb39-6a37f9c52a43           |                       |                                   |
| Group        | Information Disclosure                         | First Time Detected   | 24 Apr 2025 14:53 GMT+0630        |
| CWE          | CWE-523                                        | Last Time Detected    | 24 Apr 2025 14:53 GMT+0630        |
| OWASP        | A2 Cryptographic Failures                      | Last Time Tested      | 24 Apr 2025 14:53 GMT+0630        |
| WASC         | WASC-4 INSUFFICIENT TRANSPORT LAYER PROTECTION | Times Detected        | 1                                 |
| CVSS V3 Base | 7.5                                            | CVSS V3 Temporal      | 7.1                               |
|              |                                                | CVSS V3 Attack Vector | Network                           |

Details

Threat

An HTML form containing a password field (input field with type=password) is served insecurely over HTTP.

Impact

Serving an HTML form containing a password field (type=password) makes it susceptible to phishing attack. Failure to utilize SSL/TLS for the login landing page could allow an attacker to modify the login form action, causing the user's credentials to be posted to an arbitrary location.

Solution

Make sure that HTML forms containing password field(s) are always served over HTTPS only.

References:

- CWE-523 (Unprotected Transport of Credentials): <https://cwe.mitre.org/data/definitions/523.html>
- OWASP Transport Layer Protection Cheat Sheet (see 'Use TLS for All Pages'): [https://cheatsheetseries.owasp.org/cheatsheets/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html)

Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

Payloads (1 instance)

#1 Request

GET http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Form with credentials is served over insecure connection.  
Form action URI: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=  
  
Password fields on the form: pma\_password  
served on insecure (i.e. HTTP) page: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=

MED

150150 HTML form containing password field(s) is served over HTTP

Group IT\_Prism\_UAT

New

URL: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=%26lang=en

|              |                                                |                       |                                   |
|--------------|------------------------------------------------|-----------------------|-----------------------------------|
| Finding #    | 36276238                                       | Severity              | Confirmed Vulnerability - Level 3 |
| Unique #     | 43896670-6d01-4e66-b5ff-1e6aeb49ce97           |                       |                                   |
| Group        | Information Disclosure                         | First Time Detected   | 24 Apr 2025 14:53 GMT+0630        |
| CWE          | CWE-523                                        | Last Time Detected    | 24 Apr 2025 14:53 GMT+0630        |
| OWASP        | A2 Cryptographic Failures                      | Last Time Tested      | 24 Apr 2025 14:53 GMT+0630        |
| WASC         | WASC-4 INSUFFICIENT TRANSPORT LAYER PROTECTION | Times Detected        | 1                                 |
| CVSS V3 Base | 7.5                                            | CVSS V3 Temporal      | 7.1                               |
|              |                                                | CVSS V3 Attack Vector | Network                           |

Details

Threat

An HTML form containing a password field (input field with type=password) is served insecurely over HTTP.

Impact

Serving an HTML form containing a password field (type=password) makes it susceptible to phishing attack. Failure to utilize SSL/TLS for the login landing page could allow an attacker to modify the login form action, causing the user's credentials to be posted to an arbitrary location.

Solution

Make sure that HTML forms containing password field(s) are always served over HTTPS only.

- References:
- CWE-523 (Unprotected Transport of Credentials): <https://cwe.mitre.org/data/definitions/523.html>
  - OWASP Transport Layer Protection Cheat Sheet (see 'Use TLS for All Pages'): [https://cheatsheetseries.owasp.org/cheatsheets/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html)

Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

Payloads (1 instance)

#1 Request

GET http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Form with credentials is served over insecure connection.  
Form action URI: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=  
  
Password fields on the form: pma\_password  
served on insecure (i.e. HTTP) page: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en

MED

150150 HTML form containing password field(s) is served over HTTP

Group IT\_Prism\_UAT

New

URL: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=%26route=%2F%26token=21503a5f66643f70526c2e645b2e2956%26lang=en

|              |                                                |                       |                                   |
|--------------|------------------------------------------------|-----------------------|-----------------------------------|
| Finding #    | 36277062                                       | Severity              | Confirmed Vulnerability - Level 3 |
| Unique #     | 63e13876-9465-4e8a-b206-2f084ee496ab           |                       |                                   |
| Group        | Information Disclosure                         | First Time Detected   | 24 Apr 2025 14:53 GMT+0630        |
| CWE          | CWE-523                                        | Last Time Detected    | 24 Apr 2025 14:53 GMT+0630        |
| OWASP        | A2 Cryptographic Failures                      | Last Time Tested      | 24 Apr 2025 14:53 GMT+0630        |
| WASC         | WASC-4 INSUFFICIENT TRANSPORT LAYER PROTECTION | Times Detected        | 1                                 |
| CVSS V3 Base | 7.5                                            | CVSS V3 Temporal      | 7.1                               |
|              |                                                | CVSS V3 Attack Vector | Network                           |

## Details

### Threat

An HTML form containing a password field (input field with type=password) is served insecurely over HTTP.

### Impact

Serving an HTML form containing a password field (type=password) makes it susceptible to phishing attack. Failure to utilize SSL/TLS for the login landing page could allow an attacker to modify the login form action, causing the user's credentials to be posted to an arbitrary location.

### Solution

Make sure that HTML forms containing password field(s) are always served over HTTPS only.

#### References:

- CWE-523 (Unprotected Transport of Credentials): <https://cwe.mitre.org/data/definitions/523.html>
- OWASP Transport Layer Protection Cheat Sheet (see 'Use TLS for All Pages'): [https://cheatsheetseries.owasp.org/cheatsheets/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html)

## Detection Information

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| Parameter      | No param has been required for detecting the information.                              |
| Authentication | In order to detect this vulnerability, the scan required authentication to be enabled. |

## Payloads (2 instances)

### #1 Request

GET http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en& route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

### #1 Response

Form with credentials is served over insecure connection.  
Form action URI: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/

Password fields on the form: pma\_password  
served on insecure (i.e. HTTP) page: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en&route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329

#2 Request

GET http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&am p;token=21503a5f66643f70526c2e645b2e2956&lang=en  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*  
  
Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#2 Response

Form with credentials is served over insecure connection.  
Form action URI: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/  
  
Password fields on the form: pma\_password  
served on insecure (i.e. HTTP) page: http://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&token=21503a5f66643f70526c2e645b2e2956&lang=en

MED150263 Insecure Transport (1)

MED150263 Insecure Transport

GroupIT\_Prism\_UATNew

URL: http://prism-preprod.capgemini.com/icons/small/

|                              |                                                |                     |                                   |
|------------------------------|------------------------------------------------|---------------------|-----------------------------------|
| Finding #                    | 36277058                                       | Severity            | Confirmed Vulnerability - Level 3 |
| Unique #                     | 8617b0a5-4e55-49fb-be87-9281cf0b1e94           |                     |                                   |
| Group                        | Information Disclosure                         | First Time Detected | 24 Apr 2025 14:53 GMT+0630        |
| CWE                          | CWE-319                                        | Last Time Detected  | 24 Apr 2025 14:53 GMT+0630        |
| OWASP                        | A2 Cryptographic Failures                      | Last Time Tested    | 24 Apr 2025 14:53 GMT+0630        |
| WASC                         | WASC-4 INSUFFICIENT TRANSPORT LAYER PROTECTION | Times Detected      | 1                                 |
| CVSS V3 Base                 | 7.6                                            | CVSS V3 Temporal    | 6.6                               |
| CVSS V3 Attack VectorNetwork |                                                |                     |                                   |

Details

**Threat**  
A link is functional over an insecure, HTTP connection. No redirection to HTTPS occurs. Note that this QID is reported for 200/OK responses as well as 4xx and 5xx responses.

**Impact**  
Data sent over a non-HTTPS connection is unencrypted and vulnerable to network sniffing attacks that can expose sensitive or confidential information. This includes non-secure cookies and other potentially sensitive data contained in HTTP headers. Even if no sensitive data is transmitted, man-in-the-middle (MITM) attacks are possible over non-HTTPS connections. An attacker who exploits MITM can intercept and change the conversation between the client (e.g., web browser, mobile device, etc.) and the server.

More information: [Why HTTPS Matters](#)

**Solution**  
Ensure that all links are accessible over HTTPS only. The most secure design is for the application to listen and respond only to encrypted HTTPS requests. Alternatively, if non-HTTPS requests are accepted, the server should redirect these requests to HTTPS using a 301 or 302 response.

It is also strongly recommended to use [HTTP Strict Transport Security](#) (HSTS) so that web browsers are instructed to use only HTTPS when making requests to the server. QID 150135 will be reported when links without HSTS are found.

For more information, see the [Application section of OWASP's Transport Layer Protection Cheat Sheet](#).

Detection Information

No param has been required for detecting the information.

# WAS Web Application Report

## Parameter

**Authentication** In order to detect this vulnerability, the scan required authentication to be enabled.

## Payloads (1 instance)

### #1 Request

GET http://prism-preprod.capgemini.com/icons/small/  
Referer: https://prism-preprod.capgemini.com/  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36  
Host: prism-preprod.capgemini.com  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

### #1 Response

HTTP/1.1 200 OK  
Date: Thu, 24 Apr 2025 09:24:41 GMT  
Server: Apache  
X-Frame-Options: SAMEORIGIN  
Transfer-Encoding: chunked  
Content-Type: text/html; charset=ISO-8859-1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>Index of /icons/small</title>
</head>
<body>
<h1>Index of /icons/small</h1>
<table>
<tr><th valign="top"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M;O=A">Last modified</a></th><th><a href="?C=S;O=A">Size</a></th><th><a href="?C=D;O=A">Description</a></th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><img src="/icons/back.gif" alt="[PAREN
```

MED

## 151019 Vulnerable JavaScript Library Detected - jQuery UI (3)

MED

### 151019 Vulnerable JavaScript Library Detected - jQuery UI

Group IT\_Prism\_UAT 

New

URL: https://prism-preprod.capgemini.com/phpMyAdmin/

Finding #	36277060	Severity	Confirmed Vulnerability - Level 3
Unique #	2b4a7d90-649b-4d1c-894c-59b664c27497		
Group	Information Disclosure	First Time Detected	24 Apr 2025 14:53 GMT+0630
CWE	CWE-937	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A6 Vulnerable and Outdated Components	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	-	Times Detected	1
CVSS V3 Base	6.1	CVSS V3 Temporal	5.4
		CVSS V3 Attack Vector	Network

## Details

### Threat

jQuery UI is a collection of GUI widgets, animated visual effects, and themes implemented with jQuery (a JavaScript library), Cascading Style Sheets, and HTML.

The web application is using a JavaScript library that is known to contain at least one vulnerability.

### Impact

Attackers could potentially exploit the vulnerability in the JavaScript library. The impact of a successful exploit depends on the nature of the vulnerability and how the web application makes use of the library.



Solution

Please refer to the information provided in the response section to understand the details of the vulnerability. Check the response of the link reported to be vulnerable. If there is a redirection involved, check the response of all the intermediary links. Also check the vendor's security advisories related to the vulnerable version of the library.

Detection Information

Parameter

No param has been required for detecting the information.

Authentication

In order to detect this vulnerability, the scan required authentication to be enabled.

Access Path

Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/

Payloads (1 instance)

#1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Vulnerable javascript library: jQuery.ui.autocomplete  
version: 1.13.1  
script uri: https://prism-preprod.capgemini.com/phpMyAdmin/js/vendor/jquery/jquery-ui.min.js?v=5.2.0

Details:  
CVE-2022-31160 : jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of jQuery.  
Versions prior to 1.13.2 are potentially vulnerable to cross-site scripting. Initializing a checkboxradio widget on an input enclosed within a label makes that parent label contents considered as the input label. Calling ".checkboxradio( "refresh" )" on such a widget and the initial HTML contained encoded HTML entities will make them erroneously get decoded. This can lead to potentially executing JavaScript code.  
The bug has been patched in jQuery UI 1.13.2. To remediate the issue, someone who can change the initial HTML can wrap all the non-input contents of the "label" in a "span".

Found on the following pages (only first 10 pages are reported):  
https://prism-preprod.capgemini.com/phpMyAdmin/  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en&route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&token=21503a5f66643f70526c2e645b2e2956&lang=en  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/

MED

151019 Vulnerable JavaScript Library Detected - jQuery UI

Group IT\_Prism\_UAT

New

URL: https://prism-preprod.capgemini.com/phpMyAdmin/

Finding #	36277048	Severity	Confirmed Vulnerability - Level 3
Unique #	3d3dd104-adbc-45b9-8aac-711818b82c5c		
Group	Information Disclosure	First Time Detected	24 Apr 2025 14:53 GMT+0630
CWE	CWE-937	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A6 Vulnerable and Outdated Components	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	-	Times Detected	1
CVSS V3 Base	6.1	CVSS V3 Temporal	5.4
		CVSS V3 Attack Vector	Network

Details

Threat

# WAS Web Application Report

jQuery UI is a collection of GUI widgets, animated visual effects, and themes implemented with jQuery (a JavaScript library), Cascading Style Sheets, and HTML.

The web application is using a JavaScript library that is known to contain at least one vulnerability.

### Impact

Attackers could potentially exploit the vulnerability in the JavaScript library. The impact of a successful exploit depends on the nature of the vulnerability and how the web application makes use of the library.

### Solution

Please refer to the information provided in the response section to understand the details of the vulnerability. Check the response of the link reported to be vulnerable. If there is a redirection involved, check the response of all the intermediary links. Also check the vendor's security advisories related to the vulnerable version of the library.

Detection Information

Parameter

No param has been required for detecting the information.

Authentication

In order to detect this vulnerability, the scan required authentication to be enabled.

Access Path

Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/

Payloads (1 instance)

#1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload.Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

Vulnerable javascript library: jQuery.ui.dialog  
version: 1.13.1  
script uri: https://prism-preprod.capgemini.com/phpMyAdmin/js/vendor/jquery/jquery-ui.min.js?v=5.2.0

Details:  
CVE-2022-31160 : jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of jQuery.  
Versions prior to 1.13.2 are potentially vulnerable to cross-site scripting. Initializing a checkboxradio widget on an input enclosed within a label makes that parent label contents considered as the input label. Calling ".checkboxradio( "refresh" )" on such a widget and the initial HTML contained encoded HTML entities will make them erroneously get decoded. This can lead to potentially executing JavaScript code.  
The bug has been patched in jQuery UI 1.13.2. To remediate the issue, someone who can change the initial HTML can wrap all the non-input contents of the "label" in a "span".

Found on the following pages (only first 10 pages are reported):  
https://prism-preprod.capgemini.com/phpMyAdmin/  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en&route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&token=21503a5f66643f70526c2e645b2e2956&lang=en  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/

MED

151019 Vulnerable JavaScript Library Detected - jQuery UI

Group IT\_Prism\_UAT

New

URL: https://prism-preprod.capgemini.com/phpMyAdmin/

Finding #	36277050	Severity	Confirmed Vulnerability - Level 3
Unique #	45bad541-2cb1-41b4-a489-f00e1919d952		
Group	Information Disclosure	First Time Detected	24 Apr 2025 14:53 GMT+0630
CWE	CWE-937	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP		Last Time Tested	

# WAS Web Application Report

<a href="#">A6 Vulnerable and Outdated Components</a>				24 Apr 2025 14:53 GMT+0630
WASC	-	Times Detected		1
CVSS V3 Base	6.1	CVSS V3 Temporal	5.4	CVSS V3 Attack VectorNetwork

## Details

### Threat

jQuery UI is a collection of GUI widgets, animated visual effects, and themes implemented with jQuery (a JavaScript library), Cascading Style Sheets, and HTML.

The web application is using a JavaScript library that is known to contain at least one vulnerability.

### Impact

Attackers could potentially exploit the vulnerability in the JavaScript library. The impact of a successful exploit depends on the nature of the vulnerability and how the web application makes use of the library.

### Solution

Please refer to the information provided in the response section to understand the details of the vulnerability. Check the response of the link reported to be vulnerable. If there is a redirection involved, check the response of all the intermediary links. Also check the vendor's security advisories related to the vulnerable version of the library.

## Detection Information

Parameter	No param has been required for detecting the information.
Authentication	In order to detect this vulnerability, the scan required authentication to be enabled.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://prism-preprod.capgemini.com/

## Payloads (1 instance)

### #1 Request

GET https://prism-preprod.capgemini.com/phpMyAdmin/  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

### #1 Response

Vulnerable javascript library: jQuery.ui.tooltip  
version: 1.13.1  
script uri: https://prism-preprod.capgemini.com/phpMyAdmin/js/vendor/jquery/jquery-ui.min.js?v=5.2.0

Details:  
CVE-2022-31160 : jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of jQuery.  
Versions prior to 1.13.2 are potentially vulnerable to cross-site scripting. Initializing a checkboxradio widget on an input enclosed within a label makes that parent label contents considered as the input label. Calling ".checkboxradio( "refresh" )" on such a widget and the initial HTML contained encoded HTML entities will make them erroneously get decoded. This can lead to potentially executing JavaScript code.  
The bug has been patched in jQuery UI 1.13.2. To remediate the issue, someone who can change the initial HTML can wrap all the non-input contents of the "label" in a "span".

Found on the following pages (only first 10 pages are reported):  
https://prism-preprod.capgemini.com/phpMyAdmin/  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en&route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&token=21503a5f66643f70526c2e645b2e2956&lang=en  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en  
https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/

LOW 150112 Sensitive form field has not disabled autocomplete (1)

LOW 150112 Sensitive form field has not disabled autocomplete

Group IT\_Prism\_UAT

New

URL: https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/

Finding #	36277052	Severity	Confirmed Vulnerability - Level 2
Unique #	7296d414-c2f9-4baa-a7b5-d189a7b99013		
Group	Information Disclosure	First Time Detected	24 Apr 2025 14:53 GMT+0630
CWE	CWE-200	Last Time Detected	24 Apr 2025 14:53 GMT+0630
OWASP	A5 Security Misconfiguration	Last Time Tested	24 Apr 2025 14:53 GMT+0630
WASC	WASC-13 INFORMATION LEAKAGE	Times Detected	1
CVSS V3 Base	3.7	CVSS V3 Temporal	3.6
		CVSS V3 Attack Vector	Network

Details

**Threat**  
An HTML form that collects sensitive information does not prevent the browser from prompting the user to save the populated values for later reuse. Autocomplete should be turned off for any input that takes sensitive information such as credit card number, CVV2/CVC code, U.S. social security number, etc.

**Impact**  
If the browser is used in a shared computing environment where more than one person may use the browser, then "autocomplete" values may be submitted by an unauthorized user.

**Solution**  
Add the following attribute to the form or input element: autocomplete="off" This attribute prevents the browser from prompting the user to save the populated form values for later reuse. Most browsers no longer honor autocomplete="off" for password input fields. These browsers include Chrome, Firefox, Microsoft Edge, IE, Opera However, there is still an ability to turn off autocomplete through the browser and that is recommended for a shared computing environment. Since the ability to turn autocomplete off for password inputs fields is controlled by the user it is highly recommended for application to enforce strong password rules.

Detection Information

Parameter	No param has been required for detecting the information.
Authentication	In order to detect this vulnerability, the scan required authentication to be enabled.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:
<div>https://prism-preprod.capgemini.com/ https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/en&amp;route=%2F=en&amp;token=26693d3d43672d456d382c67715d7329</div>	

Payloads (2 instances)

#1 Request

POST https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*  
Content-Type: application/x-www-form-urlencoded

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#1 Response

The following password field(s) in the form do not set autocomplete="off":  
(Field name: pma\_password, Field id: input\_password)  
Parent URL of form is: https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en&route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329

#2 Request

POST https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/  
Host: prism-preprod.capgemini.com  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_5) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.1 Safari/605.1.15  
Accept: \*/\*  
Content-Type: application/x-www-form-urlencoded

Click this [link](#) to try to reproduce the vulnerability using above payload. Note that clicking this link may not lead to visible results, either because the vulnerability requires context to be previously set (authentication, cookies...) or because the exploitation of the vulnerability does not lead to any visible proof.

#2 Response

The following password field(s) in the form do not set autocomplete="off":  
(Field name: pma\_password, Field id: input\_password)  
Parent URL of form is: https://prism-preprod.capgemini.com/phpMyAdmin/

Information Gathered (7)

Scan Diagnostics (2)

INFO 38291 SSL Session Caching Information (1)

Group IT\_Prism\_UAT

Finding #	15287865	Severity	Information Gathered - Level 1
Unique #	b18a4169-b63c-43d1-9f21-e5145dd6fadb		
Group	Scan Diagnostics		
CWE	-	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	-		
WASC	-		

Details

Threat

SSL session is a collection of security parameters that are negotiated by the SSL client and server for each SSL connection. SSL session caching is targeted to reduce the overhead of negotiations in recurring SSL connections. SSL sessions can be reused to resume an earlier connection or to establish multiple simultaneous connections. The client suggests an SSL session to be reused by identifying the session with a Session-ID during SSL handshake. If the server finds it appropriate to reuse the session, then they both proceed to secure communication with already known security parameters.

This test determines if SSL session caching is enabled on the host.

Impact

SSL session caching is part of the SSL and TLS protocols and is not a security threat. The result of this test is for informational purposes only.

Solution

N/A

SSL Data	
Flags	-
Protocol	tcp
Virtual Host	prism-preprod.capgemini.com
IP	10.29.165.209
Port	443
Result	TLSv1.2 session caching is enabled on the target. TLSv1.3 session caching is enabled on the target.

INFO 38597 Secure Sockets Layer/Transport Layer Security (SSL/TLS) Invalid Protocol Version Tolerance (1)

INFO 38597 Secure Sockets Layer/Transport Layer Security (SSL/TLS) Invalid Protocol Version Tolerance		Group IT_Prism_UAT	
Finding #	15287867	Severity	Information Gathered - Level 1
Unique #	40fd63ef-ef57-4e44-9f47-8562fbfa5c06		
Group	Scan Diagnostics		
CWE	-	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	-		
WASC	-		

Details
---------

Threat

SSL/TLS protocols have different version that can be supported by both the client and the server. This test attempts to send invalid protocol versions to the target in order to find out what is the target's behavior. The results section contains a table that indicates what was the target's response to each of our tests.

Impact

N/A

Solution

N/A

SSL Data	
Flags	-
Protocol	tcp
Virtual Host	prism-preprod.capgemini.com
IP	10.29.165.209
Port	443
Result	#table cols=2 my_version target_version 0304 0303 0399 0303 0400 0303 0499 0303

Security Weaknesses (5)

INFO 150202 Missing header: X-Content-Type-Options (1)

INFO	150202 Missing header: X-Content-Type-Options	Group IT_Prism_UAT
------	-----------------------------------------------	--------------------

Finding #	16303337	Severity	Information Gathered - Level 2
Unique #	115e7ecb-b524-4383-bbee-1af91f5269bb		
Group	Security Weaknesses		
CWE	CWE-16, CWE-1032	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		

Details
---------

**Threat**

The X-Content-Type-Options response header is not present. WAS reports missing X-Content-Type-Options header on each crawled link for both static and dynamic responses. The scanner performs the check not only on 200 responses but 4xx and 5xx responses as well. It's also possible the QID will be reported on directory-level links.

**Impact**

All web browsers employ a content-sniffing algorithm that inspects the contents of HTTP responses and also occasionally overrides the MIME type provided by the server. If X-Content-Type-Options header is not present, browsers can potentially be tricked into treating non-HTML response as HTML. An attacker can then potentially leverage the functionality to perform a cross-site scripting (XSS) attack. This specific case is known as a Content-Sniffing XSS (CS-XSS) attack.

**Solution**

It is recommended to disable browser content sniffing by adding the X-Content-Type-Options header to the HTTP response with a value of 'nosniff'. Also, ensure that the 'Content-Type' header is set correctly on responses.

Results
---------

X-Content-Type-Options: Header missing  
Response headers on link: GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
Date: Thu, 24 Apr 2025 09:24:38 GMT  
Server: Apache  
X-Frame-Options: SAMEORIGIN  
Strict-Transport-Security: max-age=2592000; includeSubDomains  
Keep-Alive: timeout=5, max=495  
Connection: Keep-Alive  
Transfer-Encoding: chunked  
Content-Type: text/html;charset=ISO-8859-1  
Set-Cookie: S\$ESS353ac686679cf7504a2565309db31fde=2vC23nf4uupCcExyods2ng%2CrTsL21Mv3A5XsD14Kz-HrqO%2Cb; secure; HttpOnly; expires=Sat, 17-May-2025 12:57:18 GMT; domain=.prism-preprod.capgemini.com; SameSite=Lax; path=/  
Set-Cookie: PFSTG=JuRNN5Uihfbt1aKei30zN0c68PD0CAqzXlxbIenWzUmp; secure; HttpOnly; domain=.capgemini.com; SameSite=None; path=/

Header missing on the following link(s):  
(Only first 50 such pages are listed)

GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/patch.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/ response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binhex.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic3.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binhex.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/back.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/broken.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.gif response code: 200

GET https://prism-preprod.capgemini.com/icons/small/folder.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp1.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/blank.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/movie.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/key.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/burst.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/text.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image2.png response code: 200

INFO 150206 Content-Security-Policy Not Implemented (1)

INFO 150206 Content-Security-Policy Not Implemented

Group IT\_Prism\_UAT

Finding #	15319004	Severity	Information Gathered - Level 2
Unique #	15bbfc57-4325-4ef2-97a9-e038cb922faf		
Group	Security Weaknesses		
CWE	CWE-16, CWE-1032	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		

Details
---------

Threat

No Content-Security-Policy (CSP) is specified for the page. WAS checks for the missing CSP on all static and dynamic pages. It checks for CSP in the response headers (Content-Security-Policy, X-Content-Security-Policy or X-Webkit-CSP) and in response body (http-equiv="Content-Security-Policy" meta tag).

HTTP 4xx and 5xx responses can also be susceptible to attacks such as XSS. For better security it's important to set appropriate CSP policies on 4xx and 5xx responses as well.

Impact

Content-Security Policy is a defense mechanism that can significantly reduce the risk and impact of XSS attacks in modern browsers. The CSP specification provides a set of content restrictions for web resources and a mechanism for transmitting the policy from a server to a client where the policy is enforced. When a Content Security Policy is specified, a number of default behaviors in user agents are changed; specifically inline content and JavaScript eval constructs are not interpreted without additional directives. In short, CSP allows you to create a whitelist of sources of the trusted content. The CSP policy instructs the browser to only render resources from those whitelisted sources. Even though an attacker can find a security vulnerability in the application through which to inject script, the script won't match the whitelisted sources defined in the CSP policy, and therefore will not be executed.

The absence of Content Security Policy in the response will allow the attacker to exploit vulnerabilities as the protection provided by the browser is not at all leveraged by the Web application. If secure CSP configuration is not implemented, browsers will not be able to block content-injection attacks such as Cross-Site Scripting and Clickjacking.

Solution

Appropriate CSP policies help prevent content-injection attacks such as cross-site scripting (XSS) and clickjacking. It's recommended to add secure CSP policies as a part of a defense-in-depth approach for securing web applications.

References:

- [https://cheatsheetseries.owasp.org/cheatsheets/Content\\_Security\\_Policy\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html)
- <https://developers.google.com/web/fundamentals/security/csp/>



Results

Content-Security-Policy: Header missing  
Response headers on link: GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
Date: Thu, 24 Apr 2025 09:24:38 GMT  
Server: Apache  
X-Frame-Options: SAMEORIGIN  
Strict-Transport-Security: max-age=2592000; includeSubDomains  
Keep-Alive: timeout=5, max=495  
Connection: Keep-Alive  
Transfer-Encoding: chunked  
Content-Type: text/html;charset=ISO-8859-1  
Set-Cookie: S\$ESS353ac686679cf7504a2565309db31fde=2vC23nf4uupCcExyods2ng%2CrTsL21Mv3A5XsD14Kz-HrqO%2Cb; secure; HttpOnly; expires=Sat, 17-May-2025 12:57:18 GMT; domain=-prism-preprod.capgemini.com; SameSite=Lax; path=/  
Set-Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp; secure; HttpOnly; domain=-capgemini.com; SameSite=None; path=/

Header missing on the following link(s):  
(Only first 50 such pages are listed)

GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/patch.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/ response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binhex.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic3.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binhex.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/back.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/broken.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp1.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/blank.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/movie.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/key.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/burst.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/text.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image2.png response code: 200

INFO 150208 Missing header: Referrer-Policy (1)

INFO 150208 Missing header: Referrer-Policy

Group IT\_Prism\_UAT

Finding #	15319000	Severity	Information Gathered - Level 2
Unique #	ba7e3a91-5445-4982-a438-9c0af2402468		
Group			

# WAS Web Application Report

CWE	<a href="#">Security Weaknesses</a>	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	<a href="#">CWE-16, CWE-1032</a>		
WASC	<a href="#">A5 Security Misconfiguration</a>		
	<a href="#">WASC-15 APPLICATION MISCONFIGURATION</a>		

## Details

### Threat

The Referrer Policy header is used to control the flow of information from the source to the destination when a link is clicked. During the scan checks are done for the presence of the Referrer Policy on all static and dynamic pages. One of the following values for Referrer Policy in the response headers was found to be missing:

- 1) no-referrer
- 2) no-referrer-when-downgrade
- 3) same-origin
- 4) origin
- 5) origin-when-cross-origin
- 6) strict-origin
- 7) strict-origin-when-cross-origin

If the Referrer Policy header is not found, the response body is checked for a meta tag containing the tag name as "referrer" and one of the above Referrer Policy. Missing referrer header is reported for links with the following response codes - 2XX, 4xx, and 5xx. Links that report a response code of 3xx will not be tested for presence of this header.

### Impact

The Referrer-Policy header controls how much referrer information is sent to a site when navigating to it. Absence of Referrer-Policy header can lead to leakage of sensitive information via the referrer header.

### Solution

Referrer Policy header improves security by ensuring websites don't leak sensitive information via the referrer header. It's recommended to add secure Referrer Policies as a part of a defense-in-depth approach.

- References:
- <https://www.w3.org/TR/referrer-policy/>
  - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy>

## Results

Referrer-Policy: Header missing  
Response headers on link: GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
Date: Thu, 24 Apr 2025 09:24:38 GMT  
Server: Apache  
X-Frame-Options: SAMEORIGIN  
Strict-Transport-Security: max-age=2592000; includeSubDomains  
Keep-Alive: timeout=5, max=495  
Connection: Keep-Alive  
Transfer-Encoding: chunked  
Content-Type: text/html; charset=ISO-8859-1  
Set-Cookie: SSES353ac686679cf7504a2565309db31fde=2vC23nf4uupCcExyods2ng%2CrTsL21Mv3A5XsD14Kz-HrqO%2Cb; secure; HttpOnly; expires=Sat, 17-May-2025 12:57:18 GMT; domain=.prism-preprod.capgemini.com; SameSite=Lax; path=/  
Set-Cookie: PFSTG=JuRNN5Uihft1aKei30zN0c68PD0CAqzXlxbIenWzUmp; secure; HttpOnly; domain=.capgemini.com; SameSite=None; path=/

Header missing on the following link(s):  
(Only first 50 such pages are listed)

- GET https://prism-preprod.capgemini.com/icons/small/ response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/ps.png response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/patch.gif response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/transfer.gif response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/image2.gif response code: 200
- GET https://prism-preprod.capgemini.com/icons/ response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/binhex.png response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/generic3.png response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/binhex.gif response code: 200
- GET https://prism-preprod.capgemini.com/icons/small/tar.gif response code: 200

GET https://prism-preprod.capgemini.com/icons/small/comp2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/back.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/broken.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp1.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/blank.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/movie.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/key.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/burst.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/text.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image2.png response code: 200

INFO

150248 Missing header: Permissions-Policy (1)

INFO

150248 Missing header: Permissions-Policy

Group IT\_Prism\_UAT

Finding #	15319002	Severity	Information Gathered - Level 2
Unique #	9c578767-a2bc-459d-b565-6e44566c2f1d		
Group	Security Weaknesses		
CWE	CWE-284	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	A5 Security Misconfiguration		
WASC	-		

Details

Threat

The Permissions-Policy response header is not present.

Impact

Permissions-Policy allows web developers to selectively enable, disable, or modify the behavior of some of the browser features and APIs within their application.

A user agent has a set of supported features(Policy Controlled Features), which is the set of features which it allows to be controlled through policies.

Not defining policy for unused and risky policy controlled features may leave application vulnerable.

Solution

It is recommended to define policy for policy controlled features to make application more secure.

References:  
[Permissions-Policy W3C Working Draft](#)  
[Policy Controlled Features](#)

Results

Permissions-Policy: Header missing  
Response headers on link: GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
Date: Thu, 24 Apr 2025 09:24:38 GMT  
Server: Apache  
X-Frame-Options: SAMEORIGIN  
Strict-Transport-Security: max-age=2592000; includeSubDomains  
Keep-Alive: timeout=5, max=495  
Connection: Keep-Alive  
Transfer-Encoding: chunked  
Content-Type: text/html;charset=ISO-8859-1  
Set-Cookie: S\$ESS353ac686679cf7504a2565309db31fde=2vC23nf4uupCcExyods2ng%2CrTsL21Mv3A5XsD14Kz-HrqO%2Cb; secure; HttpOnly; expires=Sat, 17-May-2025 12:57:18 GMT; domain=-prism-preprod.capgemini.com; SameSite=Lax; path=/  
Set-Cookie: PFSTG=JuRNN5Uihfbt1aKei30zN0c68PD0CAqzXlxbIenWzUmp; secure; HttpOnly; domain=-capgemini.com; SameSite=None; path=/

Header missing on the following link(s):  
(Only first 50 such pages are listed)

GET https://prism-preprod.capgemini.com/icons/small/ response code: 200  
GET https://prism-preprod.capgemini.com/phpMyAdmin/ response code: 200  
GET https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&lang=en&route=%2F&lang=en&token=26693d3d43672d456d382c67715d7329 response code: 200  
GET https://prism-preprod.capgemini.com/phpMyAdmin/index.php?route=/&route=%2F&token=21503a5f66643f70526c2e645b2e2956&lang=en response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/patch.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/ response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binhex.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic3.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binhex.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/back.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/image.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/tar.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/transfer.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/broken.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder2.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/folder.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/forward.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/comp1.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/unknown.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/blank.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/index.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/sound2.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/ps.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/movie.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/key.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/binary.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/uu.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/burst.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/doc.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/continued.png response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/rainbow.gif response code: 200  
GET https://prism-preprod.capgemini.com/icons/small/generic.gif response code: 200

INFO

150135 HTTP Strict Transport Security (HSTS) header missing or misconfigured

Group IT\_Prism\_UAT

Finding #	16303335	Severity	Information Gathered - Level 1
Unique #	63f0685d-d69a-4b1b-bda8-06311eb3afd6		
Group	Security Weaknesses		
CWE	CWE-523	Detection Date	24 Apr 2025 14:53 GMT+0630
OWASP	A5 Security Misconfiguration		
WASC	-		

Details

Threat

HTTP Strict Transport Security (HSTS) header was found to be missing or misconfigured. The HSTS header instructs browsers that all subsequent connections to the website, for a configurable amount of time, should be performed over a secure (HTTPS) connection only. Additionally, it instructs browsers that users should not be permitted to bypass SSL/TLS certificate errors, in the event of an expired or otherwise untrusted certificate for example.

Impact

If HSTS header is not set, users are potentially vulnerable to man-in-the-middle (MITM) attacks, SSL stripping, and passive eavesdropper attacks.

Solution

For information about how to implement the HSTS header properly, refer to the [OWASP HTTP Strict Transport Security Cheat Sheet](#).

Results

Max-age set below 120 days for  
<https://prism-preprod.capgemini.com/icons/>






Appendix

Web Application Details  
Group IT\_Prism\_UAT

Name	Group IT_Prism_UAT
ID	833117112
URL	https://prism-preprod.capgemini.com
Scope	Limit to URL hostname
Tags	Group IT_DevSecOps, Group IT_WebApps
Custom Attributes	-

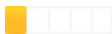



Severity Levels  
Confirmed Vulnerabilities

Vulnerabilities (QIDs) are design flaws, programming errors, or mis-configurations that make your web application and web application platform susceptible to malicious attacks. Depending on the level of the security risk, the successful exploitation of a vulnerability can vary from the disclosure of information to a complete compromise of the web application and/or the web application platform. Even if the web application isn't fully compromised, an exploited vulnerability could still lead to the web application being used to launch attacks against users of the site.

	Minimal	Basic information disclosure (e.g. web server type, programming language) might enable intruders to discover other vulnerabilities, but lack of this information does not make the vulnerability harder to find.
	Medium	Intruders may be able to collect sensitive information about the application platform, such as the precise version of software used. With this information, intruders can easily exploit known vulnerabilities specific to software versions. Other types of sensitive information might disclose a few lines of source code or hidden directories.
	Serious	Vulnerabilities at this level typically disclose security-related information that could result in misuse or an exploit. Examples include source code disclosure or transmitting authentication credentials over non-encrypted channels.
	Critical	Intruders can exploit the vulnerability to gain highly sensitive content or affect other users of the web application. Examples include certain types of cross-site scripting and SQL injection attacks.
	Urgent	Intruders can exploit the vulnerability to compromise the web application's data store, obtain information from other users' accounts, or obtain command execution on a host in the web application's architecture.

Potential Vulnerabilities

Potential Vulnerabilities indicate that the scanner observed a weakness or error that is commonly used to attack a web application, and the scanner was unable to confirm if the weakness or error could be exploited. Where possible, the QID's description and results section include information and hints for following-up with manual analysis. For example, the exploitability of a QID may be influenced by characteristics that the scanner cannot confirm, such as the web application's network architecture, or the test to confirm exploitability requires more intrusive testing than the scanner is designed to conduct.

	Minimal	Presence of this vulnerability is indicative of basic information disclosure (e.g. web server type, programming language) and might enable intruders to discover other vulnerabilities. For example in this scenario, information such as web server type, programming language, passwords or file path references can be disclosed.
	Medium	Presence of this vulnerability is indicative of basic information disclosure (e.g. web server type, programming language) and might enable intruders to discover other vulnerabilities. For example version of software or session data can be disclosed, which could be used to exploit.
	Serious	Presence of this vulnerability might give access to security-related information to intruders who are bound to misuse or exploit. Examples of what could happen if this vulnerability was exploited include bringing down the server or causing hindrance to the regular service.
	Critical	Presence of this vulnerability might give intruders the ability to gain highly sensitive content or affect other users of the web application.

# WAS Web Application Report



Urgent

Presence of this vulnerability might enable intruders to compromise the web application's data store, obtain information from other users' accounts, or obtain command execution on a host in the web application's architecture. For example in this scenario, the web application users can potentially be targeted if the application is exploited.

## Sensitive Content

Sensitive content may be detected based on known patterns (credit card numbers, social security numbers) or custom patterns (strings, regular expressions), depending on the option profile used. Intruders may gain access to sensitive content that could result in misuse or other exploits.



Minimal

Sensitive content was found in the web server response. During our scan of the site form(s) were found with field(s) for credit card number or social security number. This information disclosure could result in a confidentiality breach and could be a target for intruders. For this reason we recommend caution.



Medium

Sensitive content was found in the web server response. Specifically our service found a certain sensitive content pattern (defined in the option profile). This information disclosure could result in a confidentiality breach and could be a target for intruders. For this reason we recommend caution.

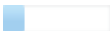


Serious

Sensitive content was found in the web server response - a valid social security number or credit card information. This information disclosure could result in a confidentiality breach, and it gives intruders access to valid sensitive content that could be misused.

## Information Gathered

Information Gathered issues (QIDs) include visible information about the web application's platform, code, or architecture. It may also include information about users of the web application.



Minimal

Intruders may be able to retrieve sensitive information related to the web application platform.



Medium

Intruders may be able to retrieve sensitive information related to internal functionality or business logic of the web application.



Serious

Intruders may be able to detect highly sensitive data, such as personally identifiable information (PII) about other users of the web application.