



report

26 Mar 2025

Vulnerabilities of all selected scans are consolidated into one report so that you can view their evolution.

PingYen Chou
tawan_pc

Chinatrust Life Insurance
Taipei
Taipei, None 115
Taiwan

Target and Filters

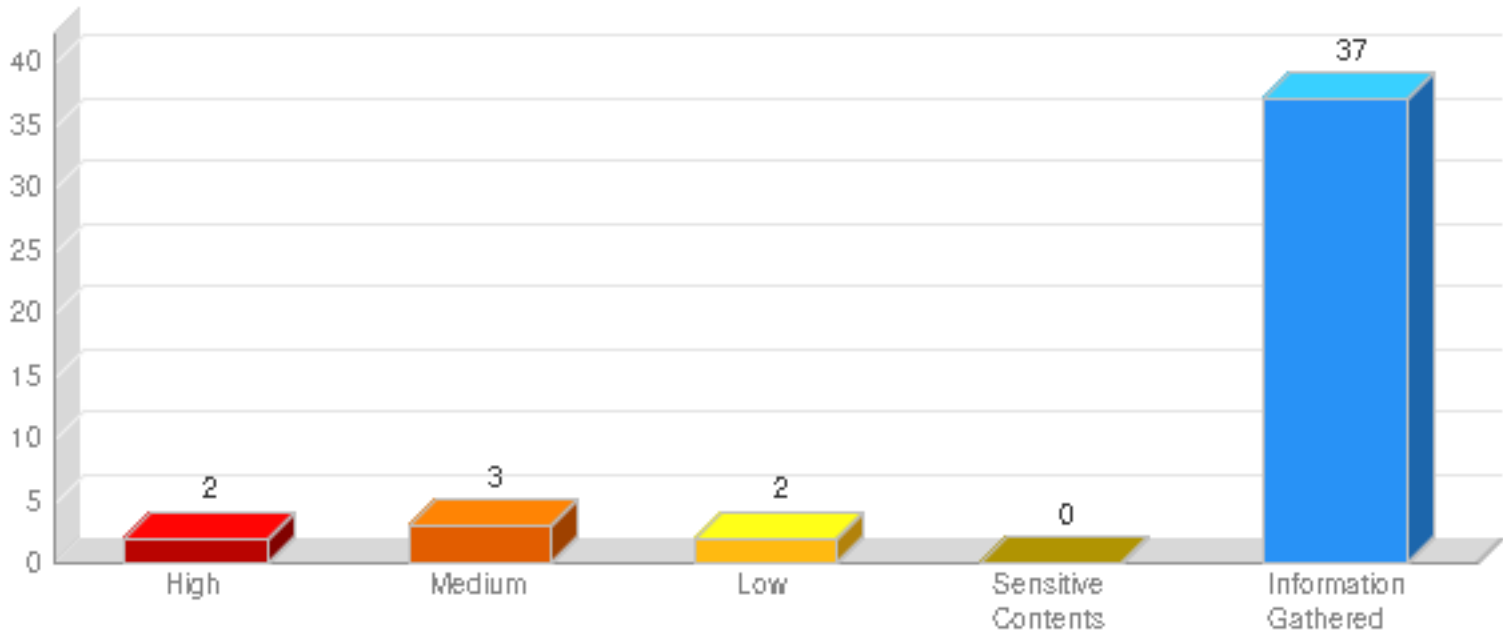
Scans (1)
Web Applications (1)

RCIS_Web Application Vulnerability Scan - Mar 25, 2025
Web地址正規化後台(RCIS_DEV)

Summary

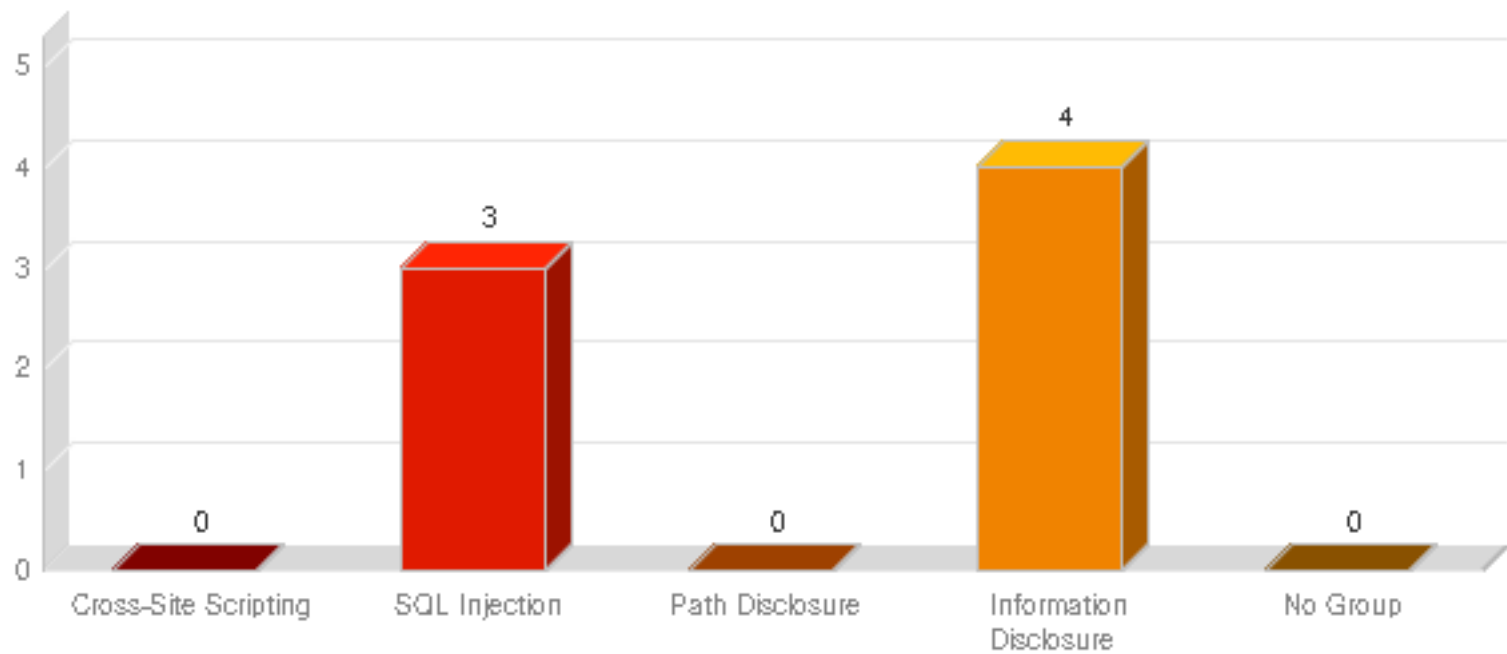
Security Risk	Vulnerabilities	Sensitive Contents	Information Gathered
HIGH	7	0	37

Findings by Severity

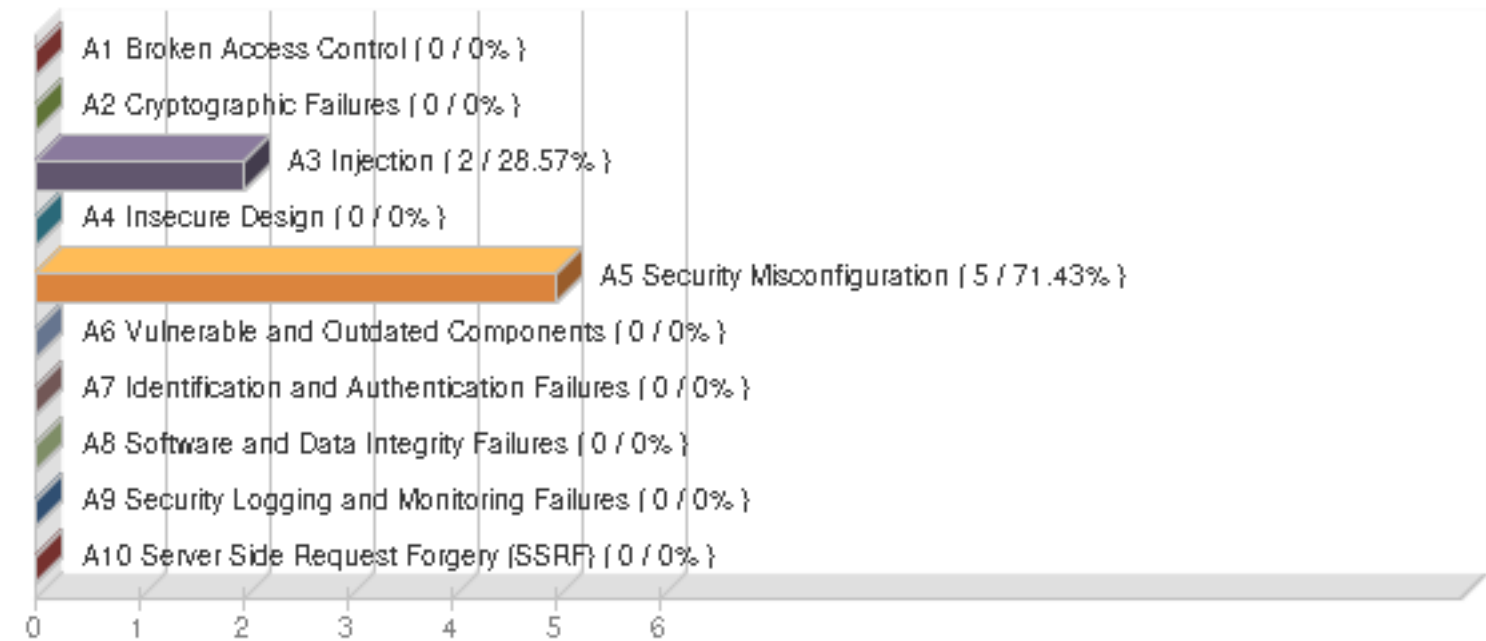


WAS Scan Report

Vulnerabilities by Group



OWASP Top 10 2021 Vulnerabilities



Scan	Date	High	Medium	Low	Sensitive Contents	Information Gathered
RCIS_Web Application Vulnerability Scan - Mar 25, 2025	25 Mar 2025 17:11 GMT +0800	2	3	2	0	37

WAS Scan Report

Results(44)

Vulnerability (7)

SQL Injection (3)

HIGH 150047 SQL Injection In HTTP Header (2)

HIGH 150047 SQL Injection In HTTP Header

URL: https://10.1.242.98/index.php/app/login

Finding #	34567792(1004328194)	Severity	Confirmed Vulnerability - Level 5
Unique #	9cceedda-f5f8-4298-b31a-47cad3e3956c		
Group	SQL Injection	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-89		
OWASP	A3 Injection		
WASC	WASC-19 SQL INJECTION		
CVSS V3 Base	10	CVSS V3 Temporal	9
CVSS V3 Attack Vector	Network		

Details

Threat

SQL injection enables an attacker to modify the syntax of a SQL query in order to retrieve, corrupt or delete data. This is accomplished by manipulating query criteria in a manner that affects the query's logic. The typical causes of this vulnerability are lack of input validation and insecure construction of the SQL query.

Queries created by concatenating strings with SQL syntax and user-supplied data are prone to this vulnerability. If any part of the string concatenation can be modified, then the meaning of the query can be changed.

In this case its payload is inside of one of the "injectable" headers of the HTTP protocol.

Examples:

These two lines demonstrate an insecure query that is created by appending the user-supplied data (**userid**):

```
dim strQuery as String
strQuery = "SELECT name,email FROM users WHERE userid=" + Request.QueryString("userid")
```

If no checks are performed against the "userid" parameter, then the query may be arbitrarily modified as shown in these two examples of a completed query:

```
SELECT name,email FROM users WHERE userid=42
SELECT name,email FROM users WHERE userid=42; SHUTDOWN WITH NOWAIT
```

Impact

The scope of a SQL injection exploit varies greatly. If any SQL statement can be injected into the query, then the attacker has the equivalent access of a database administrator. This access could lead to theft of data, malicious corruption of data, or deletion of data.

Solution

Filter all data collected from the client including browser content such as Cookies, Referrer and User-Agent headers.

SQL injection vulnerabilities can be addressed in three areas: input validation, query creation, and database security.

All input received from the Web client should be validated for correct content. If a value's type or content range is known beforehand, then stricter filters should be applied. For example, an email address should be in a specific format and only contain characters that make it a valid address; or numeric fields like a U.S. zip code should be limited to five digit values.

Prepared statements (sometimes referred to as parameterized statements) provide strong protection from SQL injection. Prepared statements are precompiled SQL queries whose parameters can be modified when the query is executed. Prepared statements enforce the logic of the query and will fail if the query cannot be compiled correctly. Programming languages that support prepared statements provide specific functions for creating queries. These functions are more secure than string concatenation for assigning user-supplied data to a query.

Stored procedures are precompiled queries that reside in the database. Like prepared statements, they also enforce separation of query data and logic. SQL statements that call stored procedures should not be created via string concatenation, otherwise their security benefits are negated.

SQL injection exploits can be mitigated by the use of Access Control Lists or role-based access within the database. For example, a read-only account would prevent an attacker from modifying data, but would not prevent the user from viewing unauthorized data. Table and row-based access controls potentially

minimize the scope of a compromise, but they do not prevent exploits.

Example of a secure query created with a prepared statement:
PreparedStatement ps = "SELECT name,email FROM users WHERE userid=?"; ps.setInt(1, userid);

Detection Information

Parameter	It has been detected by exploiting the parameter PHPSESSID The payloads section will display a list of tests that show how the param could have been exploited to collect the information
Authentication	In order to detect this vulnerability, no authentication has been required.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://10.1.242.98/

Payloads

#1 Request

GET https://10.1.242.98/index.php/app/login
Cookie: Path=/; Path=/; PHPSESSID=javascript%3Aqxss(X149196176Y3_7Z)%3B;
Referer: https://10.1.242.98/
Host: 10.1.242.98
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: Response status: 200
script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>
</body>

</html>
CDbCommand failed to execute the SQL statement: SQLSTATE[42000]: [Microsoft][ODBC Driver 18 for SQL Server][SQL Server] 'RCIS.dbo.web_session' 'id' : 'javascript:qxss(X149196176Y3_7Z)'. The SQL statement executed was: INSERT INTO [web_session] ([id], [dat

* The reflected string on the response webpage indicates that the vulnerability test was successful

HIGH 150047 SQL Injection In HTTP Header

URL: https://10.1.242.98/app/login

Finding #	35929932(1004328190)	Severity	Confirmed Vulnerability - Level 5
Unique #	074493b0-6d29-40c7-937e-cc6bb1fd4703		
Group	SQL Injection	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-89		
OWASP	A3 Injection		
WASC	WASC-19 SQL INJECTION		
CVSS V3 Base	10	CVSS V3 Temporal	9
CVSS V3 Attack Vector	Network		

Details

Threat

SQL injection enables an attacker to modify the syntax of a SQL query in order to retrieve, corrupt or delete data. This is accomplished by manipulating query criteria in a manner that affects the query's logic. The typical causes of this vulnerability are lack of input validation and insecure construction of the SQL query.

Queries created by concatenating strings with SQL syntax and user-supplied data are prone to this vulnerability. If any part of the string concatenation can be modified, then the meaning of the query can be changed.

In this case its payload is inside of one of the "injectable" headers of the HTTP protocol.

Examples:

These two lines demonstrate an insecure query that is created by appending the user-supplied data (**userid**):

```
dim strQuery as String
strQuery = "SELECT name,email FROM users WHERE userid=" + Request.QueryString("userid")
```

If no checks are performed against the "userid" parameter, then the query may be arbitrarily modified as shown in these two examples of a completed query:

```
SELECT name,email FROM users WHERE userid=42
SELECT name,email FROM users WHERE userid=42; SHUTDOWN WITH NOWAIT
```

Impact

The scope of a SQL injection exploit varies greatly. If any SQL statement can be injected into the query, then the attacker has the equivalent access of a database administrator. This access could lead to theft of data, malicious corruption of data, or deletion of data.

Solution

Filter all data collected from the client including browser content such as Cookies, Referrer and User-Agent headers.

SQL injection vulnerabilities can be addressed in three areas: input validation, query creation, and database security.

All input received from the Web client should be validated for correct content. If a value's type or content range is known beforehand, then stricter filters should be applied. For example, an email address should be in a specific format and only contain characters that make it a valid address; or numeric fields like a U.S. zip code should be limited to five digit values.

Prepared statements (sometimes referred to as parameterized statements) provide strong protection from SQL injection. Prepared statements are precompiled SQL queries whose parameters can be modified when the query is executed. Prepared statements enforce the logic of the query and will fail if the query cannot be compiled correctly. Programming languages that support prepared statements provide specific functions for creating queries. These functions are more secure than string concatenation for assigning user-supplied data to a query.

Stored procedures are precompiled queries that reside in the database. Like prepared statements, they also enforce separation of query data and logic. SQL statements that call stored procedures should not be created via string concatenation, otherwise their security benefits are negated.

SQL injection exploits can be mitigated by the use of Access Control Lists or role-based access within the database. For example, a read-only account would prevent an attacker from modifying data, but would not prevent the user from viewing unauthorized data. Table and row-based access controls potentially

minimize the scope of a compromise, but they do not prevent exploits.

Example of a secure query created with a prepared statement:
PreparedStatement ps = "SELECT name,email FROM users WHERE userid=?"; ps.setInt(1, userid);

Detection Information

Parameter	It has been detected by exploiting the parameter PHPSESSID The payloads section will display a list of tests that show how the param could have been exploited to collect the information
Authentication	In order to detect this vulnerability, no authentication has been required.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://10.1.242.98/

Payloads

#1 Request

GET https://10.1.242.98/app/login
Cookie: Path=/; Path=/; PHPSESSID=javascript%3Aqxss(X148265464Y3_7Z)%3B;
Referer: https://10.1.242.98/
Host: 10.1.242.98
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: Response status: 200
script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>
</body>

</html>
CdbCommand failed to execute the SQL statement: SQLSTATE[42000]: [Microsoft][ODBC Driver 18 for SQL Server][SQL Server] 'RCIS.dbo.web_session' 'id' : 'javascript:qxss(X148265464Y3_7Z)'. The SQL statement executed was: INSERT INTO [web_session] ([id], [dat

* The reflected string on the response webpage indicates that the vulnerability test was successful

LOW 150056 SQL Error Message (1)

LOW 150056 SQL Error Message

URL: https://10.1.242.98/api/setCookie

Finding #	34567788(1004328192)	Severity	Confirmed Vulnerability - Level 2
Unique #	3f32d109-1f3e-400a-b141-fa249047e26f		
Group	SQL Injection	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-89		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		
CVSS V3 Base	7.3	CVSS V3 Temporal	6.5
		CVSS V3 Attack Vector	Network

Details

Threat

The scan observed an SQL-based error message while performing injection tests. However, the message only appears to indicate that a SQL statement in the web application may be corrupted; it may not be exploitable.

SQL injection enables an attacker to modify the syntax of a SQL query in order to retrieve, corrupt or delete data. This is accomplished by manipulating query criteria in a manner that affects the query's logic. The typical causes of this vulnerability are lack of input validation and insecure construction of the SQL query.

Queries created by concatenating strings with SQL syntax and user-supplied data are prone to this vulnerability. If any part of the string concatenation can be modified, then the meaning of the query can be changed.

Impact

The scope of a SQL injection exploit varies greatly. If any SQL statement can be injected into the query, then the attacker has the equivalent access of a database administrator. This access could lead to theft of data, malicious corruption of data, or deletion of data.

Solution

SQL injection vulnerabilities can be addressed in three areas: input validation, query creation, and database security.

All input received from the Web client should be validated for correct content. If a value's type or content range is known beforehand, then stricter filters should be applied. For example, an email address should be in a specific format and only contain characters that make it a valid address; or numeric fields like a U.S. zip code should be limited to five digit values.

Prepared statements (sometimes referred to as parameterized statements) provide strong protection from SQL injection. Prepared statements are precompiled SQL queries whose parameters can be modified when the query is executed. Prepared statements enforce the logic of the query and will fail if the query cannot be compiled correctly. Programming languages that support prepared statements provide specific functions for creating queries. These functions are more secure than string concatenation for assigning user-supplied data to a query.

Stored procedures are precompiled queries that reside in the database. Like prepared statements, they also enforce separation of query data and logic. SQL statements that call stored procedures should not be created via string concatenation, otherwise their security benefits are negated.

SQL injection exploits can be mitigated by the use of Access Control Lists or role-based access within the database. For example, a read-only account would prevent an attacker from modifying data, but would not prevent the user from viewing unauthorized data. Table and row-based access controls potentially minimize the scope of a compromise, but they do not prevent exploits.

Example of a secure query created with a prepared statement:
PreparedStatement ps = "SELECT name,email FROM users WHERE userid=?"; ps.setInt(1, userid);

Detection Information

Parameter	It has been detected by exploiting the parameter PHPSESSID The payloads section will display a list of tests that show how the param could have been exploited to collect the information
Authentication	In order to detect this vulnerability, no authentication has been required.

WAS Scan Report

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

https://10.1.242.98/
https://10.1.242.98/index.php/app/login

Payloads

#1 Request

GET https://10.1.242.98/api/setCookie
Cookie: Path=/; PHPSESSID=javascript%3Aqxss(X152299696Y2_7Z)%3B;
Referer: https://10.1.242.98/
Host: 10.1.242.98
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: Response status: 404
script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>
</body>

</html>
CDbCommand failed to execute the SQL statement: SQLSTATE[42000]: [Microsoft][ODBC Driver 18 for SQL Server][SQL Server] 'RCIS.dbo.web_session' 'id' : 'javascript:qxss(X152299696Y2_7Z)'. The SQL statement executed was: INSERT INTO [web_session] ([id], [dat

* The reflected string on the response webpage indicates that the vulnerability test was successful

Information Disclosure (4)

MED 150022 Server Error Message (3)

MED 150022 Server Error Message

URL: https://10.1.242.98/api/setCookie

Finding #	34567790(1004328193)	Severity	Confirmed Vulnerability - Level 3
Unique #	7226c86b-78d3-4e70-b436-57b44c51e443		
Group	Information Disclosure	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-209		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		
CVSS V3 Base	5.3	CVSS V3 Temporal	4.7
		CVSS V3 Attack Vector	Network

Details

Threat

This finding will be reported when run time errors, verbose server errors, and potential stack trace are detected

The following are two examples of 150022

- WAS will send payload without URL encoding and cause an error. Note most browsers now URL encode these payloads, in order to reproduce the issue curl or a proxy can help bypass the automatic URL encoding.
- Errors can also be caused due to the speed of the scanner sending payloads. The report provides the payload that caused the error if this is consistently reported with SQL payloads it is worth taking an extra look.

Impact

Verbose error messages often expose technical details that are helpful to attackers. Verbose errors can allow attackers to learn "inside" information about the application and/or hosting infrastructure, allowing them to target it more effectively.

Solution

Implement strong error and exception handling to ensure that the web application displays only generic error messages. Avoid returning stack traces, debugging information, or other technical details to the client side. The application should also implement rigorous input data validation. Restrict user-supplied data to consist of a minimal set of characters necessary for the input field and validate the data to ensure it conforms to the expected format.

Detection Information

Parameter	It has been detected by exploiting the parameter PHPSESSID The payloads section will display a list of tests that show how the param could have been exploited to collect the information
Authentication	In order to detect this vulnerability, no authentication has been required.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:
	https://10.1.242.98/ https://10.1.242.98/index.php/app/login

Payloads

#1 Request

GET https://10.1.242.98/api/setCookie
Cookie: Path=/; PHPSESSID=javascript%3Aqxss(X152299696Y2_7Z)%3B;
Referer: https://10.1.242.98/
Host: 10.1.242.98
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: Response status: 404
script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>
</body>

</html>
CDBCommand failed to execute the SQL statement: SQLSTATE[42000]: [Microsoft][ODBC Driver 18 for SQL Server][SQL Server] 'RCIS.dbo.web_session' 'id' : 'javascript:qxss(X152299696Y2_7Z)'. The SQL statement executed was: INSERT INTO [web_session] ([id], [dat

* The reflected string on the response webpage indicates that the vulnerability test was successful

MED 150022 Server Error Message

URL: https://10.1.242.98/app/login

Finding #	35929934(1004328191)	Severity	Confirmed Vulnerability - Level 3
Unique #	5a4b660c-bb3d-4987-a3ba-de520fac43c0		
Group	Information Disclosure	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-209		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		
CVSS V3 Base	5.3	CVSS V3 Temporal	4.7
		CVSS V3 Attack Vector	Network

Details

Threat

This finding will be reported when run time errors, verbose server errors, and potential stack trace are detected

The following are two examples of 150022

- WAS will send payload without URL encoding and cause an error. Note most browsers now URL encode these payloads, in order to reproduce the issue curl or a proxy can help bypass the automatic URL encoding.
- Errors can also be caused due to the speed of the scanner sending payloads. The report provides the payload that caused the error if this is consistently reported with SQL payloads it is worth taking an extra look.

Impact

Verbose error messages often expose technical details that are helpful to attackers. Verbose errors can allow attackers to learn "inside" information about the application and/or hosting infrastructure, allowing them to target it more effectively.

Solution

Implement strong error and exception handling to ensure that the web application displays only generic error messages. Avoid returning stack traces, debugging information, or other technical details to the client side. The application should also implement rigorous input data validation. Restrict user-supplied data to consist of a minimal set of characters necessary for the input field and validate the data to ensure it conforms to the expected format.

Detection Information

Parameter	It has been detected by exploiting the parameter PHPSESSID The payloads section will display a list of tests that show how the param could have been exploited to collect the information
Authentication	In order to detect this vulnerability, no authentication has been required.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:
	https://10.1.242.98/

Payloads

#1 Request

GET https://10.1.242.98/app/login
Cookie: Path=/; Path=/; PHPSESSID=javascript%3Aqxss(X148265464Y3_7Z)%3B;
Referer: https://10.1.242.98/
Host: 10.1.242.98
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: Response status: 200
script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>
</body>

</html>
CDBCommand failed to execute the SQL statement: SQLSTATE[42000]: [Microsoft][ODBC Driver 18 for SQL Server][SQL Server] 'RCIS.dbo.web_session' 'id' : 'javascript:qxss(X148265464Y3_7Z)'. The SQL statement executed was: INSERT INTO [web_session] ([id], [dat

* The reflected string on the response webpage indicates that the vulnerability test was successful

MED 150022 Server Error Message

URL: https://10.1.242.98/index.php/app/login

Finding #	34567794(1004328195)	Severity	Confirmed Vulnerability - Level 3
Unique #	284ddf08-409a-43f3-935f-f556981d1e30		
Group	Information Disclosure	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-209		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		
CVSS V3 Base	5.3	CVSS V3 Temporal	4.7
		CVSS V3 Attack Vector	Network

Details

Threat

This finding will be reported when run time errors, verbose server errors, and potential stack trace are detected

The following are two examples of 150022

- WAS will send payload without URL encoding and cause an error. Note most browsers now URL encode these payloads, in order to reproduce the issue curl or a proxy can help bypass the automatic URL encoding.
- Errors can also be caused due to the speed of the scanner sending payloads. The report provides the payload that caused the error if this is consistently reported with SQL payloads it is worth taking an extra look.

Impact

Verbose error messages often expose technical details that are helpful to attackers. Verbose errors can allow attackers to learn "inside" information about the application and/or hosting infrastructure, allowing them to target it more effectively.

Solution

Implement strong error and exception handling to ensure that the web application displays only generic error messages. Avoid returning stack traces, debugging information, or other technical details to the client side. The application should also implement rigorous input data validation. Restrict user-supplied data to consist of a minimal set of characters necessary for the input field and validate the data to ensure it conforms to the expected format.

Detection Information

Parameter	It has been detected by exploiting the parameter PHPSESSID The payloads section will display a list of tests that show how the param could have been exploited to collect the information
Authentication	In order to detect this vulnerability, no authentication has been required.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:
	https://10.1.242.98/

Payloads

#1 Request

GET https://10.1.242.98/index.php/app/login

Cookie: Path=/; Path=/; PHPSESSID=javascript%3Aqxss(X149196176Y3_7Z)%3B;

Referer: https://10.1.242.98/

Host: 10.1.242.98

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: Response status: 200

script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>

</body>

</html>

CDBCommand failed to execute the SQL statement: SQLSTATE[42000]: [Microsoft][ODBC Driver 18 for SQL Server][SQL Server] 'RCIS.dbo.web_session' 'id' : 'javascript:qxss(X149196176Y3_7Z)'. The SQL statement executed was: INSERT INTO [web_session] ([id], [dat

* The reflected string on the response webpage indicates that the vulnerability test was successful

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

LOW 150112 Sensitive form field has not disabled autocomplete

URL: https://10.1.242.98/index.php/app/login

Finding #	34567796(1004328196)	Severity	Confirmed Vulnerability - Level 2
Unique #	89eae830-2af3-4cce-b15a-732a9dd5d176		
Group	Information Disclosure	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-200		
OWASP	A5 Security Misconfiguration		
WASC	WASC-13 INFORMATION LEAKAGE		
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, no authentication has been required.
Access Path	Here is the path followed by the scanner to reach the exploitable URL:

https://10.1.242.98/

Payloads

#1 Request

POST https://10.1.242.98/index.php/app/login
Host: 10.1.242.98
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: LoginForm[password], Field id: LoginForm_password)
Parent URL of form is: https://10.1.242.98/index.php/app/login

Information Gathered (37)

Information Disclosure (2)

INFO 150319 Weak Cookies in Use (1)

INFO 150319 Weak Cookies in Use

Finding #	15242344(1004328182)	Severity	Information Gathered - Level 2
Unique #	0e50b1ec-e9af-4b99-9006-07d634d1e74a		
Group	Information Disclosure	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-6		
OWASP	A4 Insecure Design		
WASC	-		

Details

Threat

Cookies are used to track HTTP sessions. Both session and non-session cookies could be persistent cookies in those cases it is important to verify the complexity of the cookie values.

Detection: WAS scan evaluates cookie length, analyzes for common cookie parameters not limited to PHPSESSID, ASP.NET_SessionId, JSESSIONID, sessionId, etc.

Impact

With weak cookie values, sessions can be predictable. Such cookies can be used by attacker and impersonate as a legitimate user to steal information or carry out some malicious operations.

Solution

Review cookies reported, all session cookies should have strong length, combination of alpha-number characters.

Use cryptographically secure pseudorandom number generator (CSPRNG) with a size of at least 128 bits and ensure that each sessionId is unique.

Verify non-session cookie values are strong, randomize as applicable.

Results

Weak cookies detected: 2
PHPSESSID=3100sg2cit57j55sj5ukek1v9h with issuing URI: https://10.1.242.98/, reason: Common cookie names
Path=/ with issuing URI: https://10.1.242.98/, reason: Cookie value too short

INFO 150221 External (third party) CSS link detected (1)

INFO 150221 External (third party) CSS link detected

Finding #	15713257(1004327970)	Severity	Information Gathered - Level 1
Unique #	aaa4585f-9d5b-4f2d-969a-85d2e811854e		
Group	Information Disclosure	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-830		
OWASP	-		
WASC	-		

Details

Threat

Using resources from external locations is a security concern, including third-party stylesheet. Also detection of all external resources would be a requirement for certifications and audits.

Impact

Using css from untrusted sources can result in external CSS injection and allow attacker to gain sensitive information.

Solution

Verify all the external CSS loaded on application are valid and from known sources.

Results

External CSS link found: <link href="https://fonts.googleapis.com/css2?family=Noto+Sans+TC:wght@100;300;400;500;700;900&display=swap" rel="stylesheet">
at:
https://10.1.242.98/index.php/app/login
https://10.1.242.98/api/setCookie

Scan Diagnostics (26)

INFO 45017 Operating System Detected (1)

INFO 45017 Operating System Detected

Finding #	15242335(1004327972)	Severity	Information Gathered - Level 2
Unique #	41046f90-f054-4846-b9ca-d1c975068cb2		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

Several different techniques can be used to identify the operating system (OS) running on a host. A short description of these techniques is provided below. The specific technique used to identify the OS on this host is included in the RESULTS section of your report.

1) **TCP/IP Fingerprint:** The operating system of a host can be identified from a remote system using TCP/IP fingerprinting. All underlying operating system TCP/IP stacks have subtle differences that can be seen in their responses to specially-crafted TCP packets. According to the results of this "fingerprinting" technique, the OS version is among those listed below.

Note that if one or more of these subtle differences are modified by a firewall or a packet filtering device between the scanner and the host, the fingerprinting technique may fail. Consequently, the version of the OS may not be detected correctly. If the host is behind a proxy-type firewall, the version of the operating system detected may be that of the firewall instead of the host being scanned.

2) **NetBIOS:** Short for Network Basic Input Output System, an application programming interface (API) that augments the DOS BIOS by adding special functions for local-area networks (LANs). Almost all LANs for PCs are based on the NetBIOS. Some LAN manufacturers have even extended it, adding additional network capabilities. NetBIOS relies on a message format called Server Message Block (SMB).

3) **PHP Info:** PHP is a hypertext pre-processor, an open-source, server-side, HTML-embedded scripting language used to create dynamic Web pages. Under some configurations it is possible to call PHP functions like phpinfo() and obtain operating system information.

4) **SNMP:** The Simple Network Monitoring Protocol is used to monitor hosts, routers, and the networks to which they attach. The SNMP service maintains Management Information Base (MIB), a set of variables (database) that can be fetched by Managers. These include "MIB-II.system.sysDescr" for the operating system.

Impact

Not applicable.

Solution

Not applicable.

SSL Data

Flags	-
Protocol	tcp
Virtual Host	-
IP	10.1.242.98
Port	-
Result	EulerOS_/_Ubuntu_/_Fedora_/_Tiny_Core_Linux_/_Linux_3.x_/_IBM_/_FortiSOAR_/_F5_Networks_Big-IP TCP/IP_Fingerprint M5933:7322::443

Info List

Info #1

INFO 150375 PII Fields Found (1)

INFO 150375 PII Fields Found

Finding #	15242341(1004328178)	Severity	Information Gathered - Level 2
Unique #	73255963-0397-47ac-b079-2f2401ac054c		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-359		
OWASP	A2 Cryptographic Failures		
WASC	WASC-13 INFORMATION LEAKAGE		

Details

Threat

Personally Identifiable Information(PII) is found on the form(s) on the Web Application.

Impact

Improper handling of the PII can lead to loss of reputation for the organization and the individuals whose personal information is stored. Attackers can use this information for more focused attacks in the future.

Solution

Please review all the PII fields below in the report and if required, PII should be obtained by lawful and fair means.

Results

Parent URI: https://10.1.242.98/index.php/app/login

PII fields Found:
Login Name
Password

INFO 6 DNS Host Name (1)

INFO 6 DNS Host Name

Finding #	15242327(1004327963)	Severity	Information Gathered - Level 1
Unique #	6ab6332e-36d4-4142-9a18-62a47d4890a4		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The fully qualified domain name of this host, if it was obtained from a DNS server, is displayed in the RESULT section.

Impact

N/A

Solution

N/A

SSL Data

Flags	-
Protocol	tcp
Virtual Host	-
IP	10.1.242.98
Port	-
Result	#table IP_address Host_name 10.1.242.98 No_registered_hostname

INFO 38116 SSL Server Information Retrieval (1)

INFO38116 SSL Server Information Retrieval

Finding #	15242356(1004328201)	Severity	Information Gathered - Level 1
Unique #	a5fc7db3-243a-41d5-a1d0-28a200509344		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The following is a list of supported SSL ciphers.

Note: If a cipher is included in this list it means that it was possible to establish a SSL connection using that cipher. There are some web servers setups that allow connections to be established using a LOW grade cipher, only to provide a web page stating that the URL is accessible only through a non-LOW grade cipher. In this case even though LOW grade cipher will be listed here QID 38140 will not be reported.

Impact

N/A

Solution

N/A

SSL Data

Flags	-
Protocol	tcp
Virtual Host	10.1.242.98
IP	10.1.242.98
Port	443
Result	#table cols="6" CIPHER KEY-EXCHANGE AUTHENTICATION MAC ENCRYPTION(KEY-STRENGTH) GRADE SSLv2_PROTOCOL_IS_DISABLED _ _ _ _ SSLv3_PROTOCOL_IS_DISABLED _ _ _ _ _ TLSv1_PROTOCOL_IS_DISABLED _ _ _ _ _ TLSv1.1_PROTOCOL_IS_DISABLED _ _ _ _ _ TLSv1.2_PROTOCOL_IS_ENABLED _ _ _ _ _ TLSv1.2_COMPRESSION_METHOD None _ _ _ AES128-SHA RSA RSA SHA1 AES(128) MEDIUM AES256 RSA RSA SHA1 AES(256) HIGH CAMELLIA128-SHA RSA RSA SHA1 Camellia(128) MEDIUM CAMELLIA256-SHA RSA RSA SHA1 Camellia(256) HIGH AES128-GCM-SHA256 RSA RSA AEAD AESGCM(128) MEDIUM AES256-GCM-SHA384 RSA RSA AEAD AESGCM(256) HIGH CAMELLIA128-SHA256 F RSA SHA256 Camellia(128) MEDIUM CAMELLIA256-SHA256 RSA RSA SHA256 Camellia(256) HIGH ECDHE-RSA-AES128-SHA ECDH RSA SHA1 AES MEDIUM ECDHE-RSA-AES256-SHA ECDH RSA SHA1 AES(256) HIGH ECDHE-RSA-AES128-SHA256 ECDH RSA SHA256 AES(128) MEDIUM ECDHE-AES256-SHA384 ECDH RSA SHA384 AES(256) HIGH ECDHE-RSA-AES128-GCM-SHA256 ECDH RSA AEAD AESGCM(128) MEDIUM ECDHE-RSA-AE GCM-SHA384 ECDH RSA AEAD AESGCM(256) HIGH ECDHE-RSA-CAMELLIA128-SHA256 ECDH RSA SHA256 Camellia(128) MEDIUM ECDHE-RSA-CAMELLIA256-SHA384 ECDH RSA SHA384 Camellia(256) HIGH AES128-CCM RSA RSA AEAD AESCCM(128) MEDIUM AES256-CCM RSA RSA AEAD AESCCM(256) HIGH AES128-CCM-8 RSA RSA AEAD AESCCM8(128) MEDIUM AES256-CCM-8 RSA RSA AEAD AESCCM8(256) HIGH ECDHE-RSA-CHACHA20-POLY1305 ECDH RSA AEAD CHACHA20/POLY1305(256) HIGH AES128-SHA256 RSA RSA SHA256 AES(128) MEDIUM AES256-SHA256 I RSA SHA256 AES(256) HIGH TLSv1.3_PROTOCOL_IS_DISABLED _ _ _ _ _

Info List

Info #1

Ciphers

Name	Auth	Encryption	Grade	Key Exchange	Mac	Protocol
AES128-SHA	RSA	AES(128)	MEDIUM	RSA	SHA1	TLSv1.2
AES256-SHA	RSA	AES(256)	HIGH	RSA	SHA1	TLSv1.2
CAMELLIA128-SHA	RSA	Camellia(128)	MEDIUM	RSA	SHA1	TLSv1.2
CAMELLIA256-SHA	RSA	Camellia(256)	HIGH	RSA	SHA1	TLSv1.2
AES128-GCM-SHA256	RSA	AESGCM(128)	MEDIUM	RSA	AEAD	TLSv1.2
AES256-GCM-SHA384	RSA	AESGCM(256)	HIGH	RSA	AEAD	TLSv1.2
CAMELLIA128-SHA256	RSA	Camellia(128)	MEDIUM	RSA	SHA256	TLSv1.2
CAMELLIA256-SHA256	RSA	Camellia(256)	HIGH	RSA	SHA256	TLSv1.2
ECDHE-RSA-AES128-SHA	RSA	AES(128)	MEDIUM	ECDH	SHA1	TLSv1.2
ECDHE-RSA-AES256-SHA	RSA	AES(256)	HIGH	ECDH	SHA1	TLSv1.2
ECDHE-RSA-AES128-SHA256	RSA	AES(128)	MEDIUM	ECDH	SHA256	TLSv1.2
ECDHE-RSA-AES256-SHA384	RSA	AES(256)	HIGH	ECDH	SHA384	TLSv1.2
ECDHE-RSA-AES128-GCM-SHA256	RSA	AESGCM(128)	MEDIUM	ECDH	AEAD	TLSv1.2
ECDHE-RSA-AES256-GCM-SHA384	RSA	AESGCM(256)	HIGH	ECDH	AEAD	TLSv1.2
ECDHE-RSA-CAMELLIA128-SHA256	RSA	Camellia(128)	MEDIUM	ECDH	SHA256	TLSv1.2
ECDHE-RSA-CAMELLIA256-SHA384	RSA	Camellia(256)	HIGH	ECDH	SHA384	TLSv1.2
AES128-CCM	RSA	AESCCM(128)	MEDIUM	RSA	AEAD	TLSv1.2
AES256-CCM	RSA	AESCCM(256)	HIGH	RSA	AEAD	TLSv1.2

Info List

Name	Auth	Encryption	Grade	Key Exchange	Mac	Protocol
AES128-CCM-8	RSA	AESCCM8(128)	MEDIUM	RSA	AEAD	TLSv1.2
AES256-CCM-8	RSA	AESCCM8(256)	HIGH	RSA	AEAD	TLSv1.2
ECDHE-RSA-CHACHA20-POLY1305	RSA	CHACHA20/POLY1305(256)	HIGH	ECDH	AEAD	TLSv1.2
AES128-SHA256	RSA	AES(128)	MEDIUM	RSA	SHA256	TLSv1.2
AES256-SHA256	RSA	AES(256)	HIGH	RSA	SHA256	TLSv1.2

INFO 38291 SSL Session Caching Information (1)

INFO 38291 SSL Session Caching Information

Finding #	15242353(1004328198)	Severity	Information Gathered - Level 1
Unique #	47589ac2-5352-43f2-b1f3-97dd07b86093		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
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	10.1.242.98
IP	10.1.242.98
Port	443
Result	TLSv1.2 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

Finding #	15242355(1004328200)	Severity	Information Gathered - Level 1
Unique #	0e20fa91-3a6e-4877-bf1e-9b3303be1fe0		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
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	10.1.242.98
IP	10.1.242.98
Port	443
Result	#table cols=2 my_version target_version 0304 0303 0399 0303 0400 0303 0499 0303

INFO 38704 Secure Sockets Layer/Transport Layer Security (SSL/TLS) Key Exchange Methods (1)

INFO38704 Secure Sockets Layer/Transport Layer Security (SSL/TLS) Key Exchange Methods

Finding #	15242357(1004328202)	Severity	Information Gathered - Level 1
Unique #	801aa004-7e77-4214-9aa6-580dd2ad938c		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The following is a list of SSL/TLS key exchange methods supported by the server, along with their respective key sizes, strengths and ciphers.

Impact

N/A

Solution

N/A

SSL Data

Flags	-
Protocol	tcp
Virtual Host	10.1.242.98
IP	10.1.242.98
Port	443
Result	#table cols="7" CIPHER NAME GROUP KEY-SIZE FORWARD-SECRET CLASSICAL-STRENGTH QUANTUM-STRENGTH TLSv1.2 _ _ _ _ _ AES256-SHA256 RSA _ 2048 no 110 low AES128-SHA256 RSA _ 2048 no 110 low AES256-CCM-8 RSA _ 2048 no 110 low AES128-CCM-8 RSA _ 2048 no 110 lo AES256-CCM RSA _ 2048 no 110 low AES128-CCM RSA _ 2048 no 110 low CAMELLIA256-SHA256 RSA _ 2048 no 110 low AES256-GCM-SHA384 RSA 2048 no 110 low AES128-GCM-SHA256 RSA _ 2048 no 110 low CAMELLIA256-SHA RSA _ 2048 no 110 low CAMELLIA128-SHA RSA _ 2048 no 110 low AES256-SHA RSA _ 2048 no 110 low AES128-SHA RSA _ 2048 no 110 low CAMELLIA128-SHA256 RSA _ 2048 no 110 low ECDHE-RSA-AES256-GCM-SHA384 ECDHE x25519 256 yes 128 low ECDHE-RSA-AES256-GCM-SHA384 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-AES256-GCM-SHA384 ECDHE secp256r1 256 yes 128 low ECDHE-RSA-AES256-GCM-SHA384 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-CHACHA20-POLY1305 ECDH x25519 256 yes 128 low ECDHE-RSA-CHACHA20-POLY1305 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-CHACHA20-POLY1305 ECDHE secp256 yes 128 low ECDHE-RSA-CHACHA20-POLY1305 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-AES128-GCM-SHA256 ECDHE x25519 256 yes 128 low ECDHE-RSA-AES128-GCM-SHA256 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-AES128-GCM-SHA256 ECDHE secp256r1 256 yes 128 low ECDH RSA-AES128-GCM-SHA256 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-AES256-SHA384 ECDHE x25519 256 yes 128 low ECDHE-RSA-AES256-SHA384 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-AES256-SHA384 ECDHE secp256r1 256 yes 128 low ECDHE-RSA-AES256-SHA384 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-CAMELLIA256-SHA384 ECDHE x25519 256 yes 128 low ECDHE-RSA-CAMELLIA256-SHA384 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-CAMELLIA256-SHA384 ECDHE secp256r1 256 yes 128 low ECDHE-RSA-CAMELLIA256-SHA384 ECDHE secp521r1 521 y 260 low ECDHE-RSA-AES128-SHA256 ECDHE x25519 256 yes 128 low ECDHE-RSA-AES128-SHA256 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-AES128-SHA256 ECDHE secp256r1 256 yes 128 low ECDHE-RSA-AES128-SHA256 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-CAMELLIA128-SHA256 ECDHE x25519 256 yes 128 low ECDHE-RSA-CAMELLIA128-SHA256 ECDHE secp384r1 384 yes 192 low ECDHE-RSA-CAMELLIA128-SHA256 ECDHE secp256r1 256 yes 128 low ECDHE-RSA-CAMELLIA128-SHA256 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-AES256-SHA ECDHE x25519 256 yes 128 low ECDHE-RSA-AES256-SHA ECDHE secp384r1 384 yes 192 low ECDHE-RSA-AES256-SHA ECDHE secp256r1 256 yes 128 low ECDHE-RSA-AES256 ECDHE secp521r1 521 yes 260 low ECDHE-RSA-AES128-SHA ECDHE x25519 256 yes 128 low ECDHE-RSA-AES128-SHA ECDHE secp384r1 384 yes 1 low ECDHE-RSA-AES128-SHA ECDHE secp256r1 256 yes 128 low ECDHE-RSA-AES128-SHA ECDHE secp521r1 521 yes 260 low

Info List

Info #1

Kexs

Kex	Group	Protocol	Key Size	Fwd Sec	Classical	Quantam
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
RSA		TLSv1.2	2048	no	110	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low

Info List

Kex	Group	Protocol	Key Size	Fwd Sec	Classical	Quantam
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low

Info List

Kex	Group	Protocol	Key Size	Fwd Sec	Classical	Quantam
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	384	yes	192	low
ECDHE		TLSv1.2	256	yes	128	low
ECDHE		TLSv1.2	521	yes	260	low

INFO 38706 Secure Sockets Layer/Transport Layer Security (SSL/TLS) Protocol Properties (1)

INFO

38706 Secure Sockets Layer/Transport Layer Security (SSL/TLS) Protocol Properties

Finding #	15242358(1004328203)	Severity	Information Gathered - Level 1
Unique #	c62c3b27-bd0a-4285-8873-9f53fa9a41a7		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The following is a list of detected SSL/TLS protocol properties.

Impact

- Items include:
- Extended Master Secret: indicates whether the extended_master_secret extension is supported or required by the server. This extension enhances security and is recommended. Applicable to TLSv1, TLSv1.1, TLSv1.2, DTLSv1, DTLSv1.2
 - Encrypt Then MAC: indicates whether the encrypt_then_mac extension is supported or required by the server. This extension enhances the security of non-AEAD ciphers and is recommended. Applicable to TLSv1, TLSv1.1, TLSv1.2, DTLSv1, DTLSv1.2
 - Heartbeat: indicates whether the heartbeat extension is supported. It is not recommended to enable this, except for DTLS. Applicable to TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, DTLSv1, DTLSv1.2
 - Truncated HMAC: indicates whether the truncated_hmac extension is supported. This can degrade security and is not recommended. Applicable to TLSv1, TLSv1.1, TLSv1.2, DTLSv1, DTLSv1.2
 - Cipher priority: indicates whether client, server or both determine the priority of ciphers. Having the server determine the priority is recommended. Applicable to SSLv3, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3, DTLSv1, DTLSv1.2

Solution

N/A

SSL Data

Flags	-
Protocol	tcp
Virtual Host	10.1.242.98
IP	10.1.242.98
Port	443
Result	#table cols="2" NAME STATUS TLSv1.2 _ Extended_Master_Secret yes Encrypt_Then_MAC yes Heartbeat no Truncated_HMAC no Cipher_priority_controlled_by client OCSP_stapling no SCT_extension no

Info List

Info #1

Props

Name	Value	Protocol
Extended Master Secret	yes	TLSv1.2
Encrypt Then MAC	yes	TLSv1.2
Heartbeat	no	TLSv1.2
Truncated HMAC	no	TLSv1.2
Cipher priority controlled by	client	TLSv1.2
OCSP stapling	no	TLSv1.2
SCT extension	no	TLSv1.2

INFO 42350 TLS Secure Renegotiation Extension Support Information (1)

INFO 42350 TLS Secure Renegotiation Extension Support Information

Finding #	15242354(1004328199)	Severity	Information Gathered - Level 1
Unique #	362dbb29-b2dd-4847-b2f3-b242d3aee296		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

Secure Socket Layer (SSL) and Transport Layer Security (TLS) renegotiation are vulnerable to an attack in which the attacker forms a TLS connection with the target server, injects content of his choice, and then splices in a new TLS connection from a client. The server treats the client's initial TLS handshake as a renegotiation and thus believes that the initial data transmitted by the attacker is from the same entity as the subsequent client data. TLS protocol was extended to cryptographically tierenegotiations to the TLS connections they are being performed over. This is referred to as TLS secure renegotiation extension. This detection determines whether the TLS secure renegotiation extension is supported by the server or not.

Impact

N/A

Solution

N/A

SSL Data

Flags	-
Protocol	tcp
Virtual Host	10.1.242.98
IP	10.1.242.98
Port	443
Result	TLS Secure Renegotiation Extension Status: supported.

INFO 45038 Host Scan Time - Scanner (1)

INFO 45038 Host Scan Time - Scanner

Finding #	15242345(1004328183)	Severity	Information Gathered - Level 1
Unique #	78fbc39f-e3d4-4d99-9372-79a65b02fd55		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The Host Scan Time is the period of time it takes the scanning engine to perform the vulnerability assessment of a single target host. The Host Scan Time for this host is reported in the Result section below.

The Host Scan Time does not have a direct correlation to the Duration time as displayed in the Report Summary section of a scan results report. The Duration is the period of time it takes the service to perform a scan task. The Duration includes the time it takes the service to scan all hosts, which may involve parallel scanning. It also includes the time it takes for a scanner appliance to pick up the scan task and transfer the results back to the service's Secure Operating Center. Further, when a scan task is distributed across multiple scanners, the Duration includes the time it takes to perform parallel host scanning on all scanners.

Impact

N/A

Solution

N/A

Results

Scan duration: 5260 seconds

Start time: Tue, Mar 25 2025, 09:11:30 GMT

End time: Tue, Mar 25 2025, 10:39:10 GMT

INFO 86002 SSL Certificate - Information (1)

INFO86002 SSL Certificate - Information

Finding #	15242351(1004328197)	Severity	Information Gathered - Level 1
Unique #	1e863057-3458-48a1-9bb0-9b5ea2466efd		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

SSL certificate information is provided in the Results section.

Impact

N/A

Solution

N/A

SSL Data

Flags	-
Protocol	tcp
Virtual Host	10.1.242.98
IP	10.1.242.98
Port	443
Result	#table cols="2" NAME VALUE (0)CERTIFICATE_0_(0)Version 3_(0x2) (0)Serial_Number_47:e8:00:00:00:07:8a:7b:16:be:bf:5b:c4:d5:a9:23_(0)Signature_Algorithm sha256WithRSAEncryption (0)ISSUER_NAME_countryName TW_organizationName TAIWAN-CA_commonName TWCA_Secure_SSL_Certification_Authority (0)SUBJECT_NAME_countryName TW_stateOrProvinceName Taiwan_localityName Taipei_organizationName Taiwan_Life_Insurance_Co.,_Ltd._commonName *.taiwanlife.com (0)Valid_From Oct_22_07:49:49_2024_GMT (0)Valid_Till Nov_21_15:59:59_2025_GMT (0)Public_Key_Algorithm rsaEncryption (0)RSA_Public_Key (2048_bit) (0)_RSA_Public-Key:_(2048_bit) (0)_Modulus: (0)_00:f6:8b:02:0d:82:e3:f3:25:14:30:8:47:29: (0)_06:31:ea:78:c6:bb:68:a1:e5:aa:a6:ee:28:d2:62: (0)_be:cf:c3:16:4d:7a:57:c8:10:6f:7b:c1:fd:77:a4: (0)_18:af:6c:59:e3:63:84:d7:0e:5f:ea:08:bb:39:79: (0)_6a:60:af:d1:17:86:1a:dc:43:11:5b:66:d6:14:e4: (0)_cf:db:8d:c3:c4:95:e2:6a:b1:cf:df:71:d3:e3:25: (0)_f1:3e:78:12:2a:f8:7f:b9:7b:59:b4:cc:77:c6:e0:f9:58:80:a1:b6:7b:7e:dc:7c:19:fe:55:8f:8e: (0)_9a:48:bc:59:f7:f3:13:74:a1:bf:a7:73:b1:79:6f: (0)_c5:d9:79:a7:3b:41:45:40:00:8b:b3:f8:a8:78:d6: (0)_9e:f5:56:43:30:b1:cd:d4:61:5e:ad:b5:d2:7e:bb: (0)_79:a3:1c:52:4b:58:8a:1b:df:de:7f:05:17:dd:4b: (0)_2c:37:bc:eb:20:06:c8:ab:6a:81:d0:81:db:52:55: (0)_95:38:c6:6a:1d:25:a1:f6:85:4a:53:a0:69:30: (0)_d5:56:5b:28:bd:4c:73:5c:5c:aa:5e:10:58:8e:8f: (0)_5e:73:5e:30:eb:d3:60:32:00:cd:04:08:48:ce:03: (0)_3a:04:91:c6:2c:d5:c1:5c:d1:15:ee:a2:e6:7f:d2: (0)_b9:5d (0)_Exponent: 65537_(0x10001) (0)X509v3_EXTENSIONS_(0)X509v3_Authority_Key_Identifier_1:92:E7:FA:62:16:71:8C:F3:97:71:42:C6:06:A7:E0:46:61:4B:5C:B6 (0)X509v3_Subject_Key_Identifier_92:A6:DA:65:6A:5A:18:7A:0B:C4:ED:48:A1:89:2E:BB:7C:D5:BF:58:5B:51:49:AF:A4:9A:62:5F:2D:49:15:B4 (0)X509v3_CRL_Distribution_Points (0)_Full_Name: (0)_URI:http://sslserver.twca.com.tw/sslserver/Securessl_revoke_sha2_2023G3.crl (0)X509v3_Subject_Alternative_Name_DNS:*.taiwanlife.com_DNS:taiwanlife.com (0)Authority_Information_Access_CA_Issuers_-_URI:http://sslserver.twca.com.tw/cacert/secure_sha2_2023G3.crt (0)_OCSP_-_URI:http://twcasslocsp.twca.com.tw/ (0)X509v3_Certificate_Policies_Policy: 1.3.6.1.4.1.40869.1.1.21 (0)_CPS: https://www.twca.com.tw/ (0)_Policy: 2.23.140.1.2.2 (0)X509v3_Basic_Constraints critical (0)_CA:FALSE (0)X509v3_Key_Usage critical (0)_Digital_Signature_Key_Encipherment (0)X509v3_Extended_Key_Usage_TLS_Web_Server_Authentication_TLS_Web_Client_Authentication (0)CT_Precertificate_SCTs_Signed_Certificate_Timestamp: (0)_Version:_v1_(0x0) _Log_ID_: E6:D2:31:63:40:77:8C:C1:10:41:06:D7:71:B9:CE:C1: (0)_D2:40:F6:96:84:86:FB:BA:87:32:1D:FD:1E:37:8E:50 (0)_Timestamp_: Oct_22_07:49:50.655_2024_GMT (0)_Extensions: none (0)_Signature_: ecdsa-with-SHA256 (0)_30:44:02:20:2B:CD:E2:9A:C2:C6:40:57:24:94:43:C8: (0)_8E:E8:9A:C8:F6:B1:9D:0F:30:75:EB:D8:63:CC:E7:8A: (0)_9D:64:E0:B4:02:20:53:BC:B9:DB:D0:DA:AC:63:FB:E0: (0)_22:F4:A5:8A:16:0C:BA:98:1C:18:D6:1B:E9:A5:6E:8F: (0)_C1:FB:AD:E2:29:B3 (0)_Signed_Certificate_Timestamp: (0)_Version:_v1_(0x0) (0)_Log_ID_: 12:F1:4E:34:BD:53:72:4C:84:06:19:C3:8F:3F:7A:13: (0)_F8:E7:B5:62:87:88:9C:61:30:05:84:EB:E5:86:26:3A (0)_Timestamp_: Oct_22_07:49:50.877_2024_GMT (0)_Extensions: none (0)_Signature_: ecdsa-with-SHA256 (0)_30:45:02:14:C6:75:74:5F:99:D5:61:13:7B:21: (0)_19:65:C8:A4:BF:8F:76:6A:A6:FC:30:E4:C7:DE:EA:97: (0)_F7:ED:5D:0F:02:21:00:FD:8C:D0:F2:C8:D9:FD:60:CC: _6D:71:C5:E8:3F:82:F7:78:B2:E8:63:67:F5:AD:99:CC: (0)_B3:DE:6A:8E:84:DE:9E (0)_Signed_Certificate_Timestamp: (0)_Version:_v1_(0x0) (0)_Log_ID_: 28:2C:8B:DD:81:0F:F9:09:12:0A:CE:16:D6:E0:EC:20: (0)_1B:EA:82:A3:A4:AF:19:D9:EF:FB:59:E8:3F:DC:42:68 (0)_Timestamp_: Oct_22_07:49:50.277_2024_GMT (0)_Extensions: none (0)_Signature_: ecdsa-with-SHA256 (0)_30:44:02:20:14:C8:3F:90:CB:7A:6D:56:97:31:F4:C2: (0)_5B:F6:42:6A:85:4C:BE:8C:1D:BB:72:6B:04:D1:E5:7A: (0)_52:AD:1E:5A:02:20:71:81:76:7C:86:F2:30:5B:B0:68: (0)_45:BE:52:79:E2:67:F:A0:14:92:3D:AA:85:7C:66: (0)_3C:34:7F:54:9C:71 (0)Signature (256_octets) (0) 96:90:9e:13:43:b5:5b:99:1e:7a:4f:bb:ab:7c:5d:30 (0) b6:d8:d9:20:b1:15:54:0e:bf:75:0f:c0:76:90:f8:20 (0) 36:79:4c:2d:b9:30:72:3b:1c:2a:e6:3a:18:f1:36:66 (0) 44:38:b5:62:29:0b:2e:87:e9:15:c3:26:04:98:43:dd ((6b:85:eb:73:7a:e7:8e:4e:6b:76:23:cd:80:db:ef (0) df:28:f3:58:17:11:94:1c:1d:33:e4:25:27:45:6b:66 (0) f4:64:c0:54:1c:86:37:11:f1:87:4b:e4:54:fb:d1:a0 (0) 69:eb:a9:43:c7:04:73:19:1a:bc:0c:2e:4a:83:41:8c (0) a5:39:30:33:20:8f:b5:98:64:a1:31:af:73:2b:1a:89 (0) a3:52:86:aa:66:b2:9f:ac:26:49:0a:24:10:9c:3a:6e ((ae:a7:33:32:66:76:ac:5a:4b:06:00:cc:d2:47:ec:7c (0) 3a:9a:ae:35:44:9d:51:1e:f0:28:ba:23:c2:53:cc:30 (0) ef:f1:03:5a:14:dd:e5:6e:4f:fa:d8:8b:c4:c3:12:4e (0) d3:a8:d4:58:2a:f0:0c:a5:c6:29:86:3f:4e:3a:0c:ae (0) 35:90:f0:62:7a:a2:72:26:c5:6e:28:6a:5f:6e:4d:75 (0) 27:23:aa:ca:c6:0c:b2:d2:b9:b0:0c:b6:51:44:6d:cb

Info List

Info #1

Certificate Fingerprint:BC7B4876BE554DDE79D7BF0A769CC45A5FCFA48BB7B93277ADBBAB3B6222C903

INFO 150009 Links Crawled (1)

INFO 150009 Links Crawled

Finding #	15242349(1004328187)	Severity	Information Gathered - Level 1
Unique #	337380c1-f5a2-42ab-a398-200d4bd207a4		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The list of unique links crawled and HTML forms submitted by the scanner appear in the Results section. This list may contain fewer links than the maximum threshold defined.

- NOTE: This list also includes:
- All the unique links that are reported in QID 150140 (Redundant links/URL paths crawled and not crawled)
 - All the forms reported in QID 150152 (Forms Crawled)
 - All the forms in QID 150115 (Authentication Form Found)
 - Certain requests from QID 150172 (Requests Crawled)

Impact

N/A

Solution

N/A

Results

Duration of crawl phase (seconds): 2048.00
Number of links: 9
(This number excludes form requests, ajax links (included in QID 150148) and links re-requested during authentication.)

https://10.1.242.98/
https://10.1.242.98/api/setCookie
https://10.1.242.98/app/index
https://10.1.242.98/app/login
https://10.1.242.98/index.php/app/login
https://10.1.242.98/protected/views/app/images/background02.png
https://10.1.242.98/resources/30d9a0ac/images/icon/favicon.png
https://10.1.242.98/resources/4ef8da87/font/Noto_Sans_TC/NotoSansTC-Regular.otf
https://10.1.242.98/resources/4ef8da87/images/errorBac.png

INFO 150010 External Links Discovered (1)

WAS Scan Report

INFO

150010 External Links Discovered

Finding #	15242347(1004328185)	Severity	Information Gathered - Level 1
Unique #	8790c38e-069f-4031-8061-bc3f539715ab		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

External links discovered during the scan are listed in the Results section. These links were out of scope for the scan and were not crawled.

Impact

N/A

Solution

N/A

Results

Number of links: 6
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js
https://fonts.gstatic.com/
https://fonts.googleapis.com/
https://fonts.googleapis.com/css2?family=Noto+Sans+TC:wght@100;300;400;500;700;900&display=swap
https://rcisuat.taiwanlife.com/.
https://rcisuat.taiwanlife.com/protected/views/app/images/.

INFO

150020 Links Rejected By Crawl Scope or Exclusion List (1)

INFO 150020 Links Rejected By Crawl Scope or Exclusion List

Finding #	15242328(1004327964)	Severity	Information Gathered - Level 1
Unique #	d63f9a43-dfad-4bd2-9964-d10d43603497		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

One or more links were not crawled because of an explicit rule to exclude them. This also occurs if a link is malformed.

Exclude list and Include list entries can cause links to be rejected. If a scan is limited to a specific starting directory, then links outside that directory will neither be crawled or tested.

Links that contain a host name or IP address different from the target application are considered external links and not crawled by default; those types of links are not listed here. This often happens when the scope of a scan is limited to the directory of the starting URL. The scope can be changed in the Web Application Record.

During the test phase, some path-based tests may be rejected if the scan is limited to the directory of the starting URL and the test would fall outside that directory. In these cases, the number of rejected links may be too high to list in the Results section.

Impact

Links listed here were neither crawled or tested by the Web application scanning engine.

Solution

A link might have been intentionally matched by a exclude or include list entry. Verify that no links in this list were unintentionally rejected.

Results

Links not permitted:
(This list includes links from QIDs: 150010,150041,150143,150170)

External links discovered:
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js
https://fonts.gstatic.com/
https://fonts.googleapis.com/
https://fonts.googleapis.com/css2?family=Noto+Sans+TC:wght@100;300;400;500;700;900&display=swap
https://rcisuat.taiwanlife.com/.
https://rcisuat.taiwanlife.com/protected/views/app/images/.

IP based excluded links:
Links rejected during the test phase not reported due to volume of links.

INFO 150021 Scan Diagnostics (1)

INFO 150021 Scan Diagnostics

Finding #	15242330(1004327966)	Severity	Information Gathered - Level 1
Unique #	c4be4681-3bc5-4e96-9ff2-02b59beb6c26		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

This check provides various details of the scan's performance and behavior. In some cases, this check can be used to identify problems that the scanner encountered when crawling the target Web application.

Impact

The scan diagnostics data provides technical details about the crawler's performance and behavior. This information does not necessarily imply problems with the Web application.

Solution

No action is required.

Results

Loaded 0 exclude list entries.
Loaded 0 allow list entries.
HTML form authentication unavailable, no WEBAPP entry found
Target web application page https://10.1.242.98/ fetched. Status code:302, Content-Type:text/html, load time:1 milliseconds.
Batch #0 VirtualHostDiscovery: estimated time < 1 minute (0 tests, 0 inputs)
VirtualHostDiscovery: 0 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #0 SameSiteScripting: estimated time < 1 minute (0 tests, 0 inputs)
SameSiteScripting: 0 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #0 CMSDetection: estimated time < 1 minute (1 tests, 1 inputs)
[CMSDetection phase] : No potential CMS found using Blind Elephant algorithm. Aborting the CMS Detection phase
CMSDetection: 1 vulnsigs tests, completed 56 requests, 16 seconds. Completed 56 requests of 56 estimated requests (100%). All tests completed.
Collected 19 links overall in 0 hours 34 minutes duration.
Batch #0 BannersVersionReporting: estimated time < 1 minute (1 tests, 1 inputs)
BannersVersionReporting: 1 vulnsigs tests, completed 0 requests, 0 seconds. Completed 0 requests of 1 estimated requests (0%). All tests completed.
WS Directory Path manipulation no tests enabled.
Batch #0 WS enumeration: estimated time < 1 minute (11 tests, 25 inputs)
WS enumeration: 11 vulnsigs tests, completed 195 requests, 8 seconds. Completed 195 requests of 275 estimated requests (70.9091%). All tests completed.
Batch #1 URI parameter manipulation (no auth): estimated time < 1 minute (146 tests, 0 inputs)
Batch #1 URI parameter manipulation (no auth): 146 vulnsigs tests, completed 0 requests, 14 seconds. No tests to execute.
Batch #1 Form parameter manipulation (no auth): estimated time < 1 minute (146 tests, 3 inputs)
Batch #1 Form parameter manipulation (no auth): 146 vulnsigs tests, completed 426 requests, 78 seconds. Completed 426 requests of 438 estimated requests (97.2603%). All tests completed.
Batch #1 Potential SSRF Detection URI parameter manipulation (no auth): estimated time < 1 minute (146 tests, 0 inputs)
Batch #1 Potential SSRF Detection URI parameter manipulation (no auth): 146 vulnsigs tests, completed 0 requests, 13 seconds. No tests to execute.
Batch #1 Potential SSRF Detection Form parameter manipulation (no auth): estimated time < 1 minute (146 tests, 3 inputs)
Batch #1 Potential SSRF Detection Form parameter manipulation (no auth): 146 vulnsigs tests, completed 12 requests, 17 seconds. Completed 12 requests of 438 estimated requests (2.73973%). All tests completed.
Batch #1 URI blind SQL manipulation (no auth): estimated time < 1 minute (13 tests, 0 inputs)
Batch #1 URI blind SQL manipulation (no auth): 13 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #1 Form blind SQL manipulation (no auth): estimated time < 1 minute (13 tests, 3 inputs)
Batch #1 Form blind SQL manipulation (no auth): 13 vulnsigs tests, completed 72 requests, 13 seconds. Completed 72 requests of 117 estimated requests (61.5385%). All tests completed.
Batch #1 URI parameter time-based tests (no auth): estimated time < 1 minute (21 tests, 0 inputs)
Batch #1 URI parameter time-based tests (no auth): 21 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #1 Form field time-based tests (no auth): estimated time < 1 minute (21 tests, 3 inputs)
Batch #1 Form field time-based tests (no auth): 21 vulnsigs tests, completed 63 requests, 11 seconds. Completed 63 requests of 63 estimated requests (100%). All tests completed.
Batch #1 URI parameter time-based tests for Apache Struts Vulnerabilities (no auth): estimated time < 1 minute (1 tests, 0 inputs)
Batch #1 URI parameter time-based tests for Apache Struts Vulnerabilities (no auth): 1 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #1 Form field time-based tests for Apache Struts Vulnerabilities (no auth): estimated time < 1 minute (1 tests, 3 inputs)
Batch #1 Form field time-based tests for Apache Struts Vulnerabilities (no auth): 1 vulnsigs tests, completed 3 requests, 0 seconds. Completed 3 requests of 3 estimated requests (100%). All tests completed.
Batch #2 URI parameter manipulation (no auth): estimated time < 1 minute (146 tests, 0 inputs)
Batch #2 URI parameter manipulation (no auth): 146 vulnsigs tests, completed 0 requests, 14 seconds. No tests to execute.
Batch #2 Potential SSRF Detection URI parameter manipulation (no auth): estimated time < 1 minute (146 tests, 0 inputs)
Batch #2 Potential SSRF Detection URI parameter manipulation (no auth): 146 vulnsigs tests, completed 0 requests, 14 seconds. No tests to execute.
Batch #2 URI blind SQL manipulation (no auth): estimated time < 1 minute (13 tests, 0 inputs)
Batch #2 URI blind SQL manipulation (no auth): 13 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #2 URI parameter time-based tests (no auth): estimated time < 1 minute (21 tests, 0 inputs)
Batch #2 URI parameter time-based tests (no auth): 21 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.
Batch #2 URI parameter time-based tests for Apache Struts Vulnerabilities (no auth): estimated time < 1 minute (1 tests, 0 inputs)

WAS Scan Report

Batch #2 URI parameter time-based tests for Apache Struts Vulnerabilities (no auth): 1 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.

Batch #4 WebCgiOob: estimated time < 10 minutes (161 tests, 1 inputs)

Batch #4 WebCgiOob: 161 vulnsigs tests, completed 1226 requests, 105 seconds. Completed 1226 requests of 5050 estimated requests (24.2772%). All tests completed.

Batch #4 Potential LDAP Login Bypass Tests: estimated time < 1 minute (146 tests, 3 inputs)

Batch #4 Potential LDAP Login Bypass Tests: 146 vulnsigs tests, completed 2 requests, 1262 seconds. Completed 2 requests of 438 estimated requests (0.456621%). All tests completed.

No XML requests found. Skipping XXE tests.

Batch #4 DOM XSS exploitation: estimated time < 1 minute (4 tests, 0 inputs)

Batch #4 DOM XSS exploitation: 4 vulnsigs tests, completed 0 requests, 1 seconds. No tests to execute.

Batch #4 HTTP call manipulation: estimated time < 1 minute (38 tests, 0 inputs)

Batch #4 HTTP call manipulation: 38 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.

Batch #4 Open Redirect analysis: estimated time < 1 minute (2 tests, 2 inputs)

Batch #4 Open Redirect analysis: 2 vulnsigs tests, completed 4 requests, 498 seconds. Completed 4 requests of 4 estimated requests (100%). All tests completed.

CSRF tests will not be launched because the scan is not successfully authenticated.

Batch #4 File Inclusion analysis: estimated time < 1 minute (1 tests, 12 inputs)

Batch #4 File Inclusion analysis: 1 vulnsigs tests, completed 0 requests, 0 seconds. Completed 0 requests of 12 estimated requests (0%). All tests completed.

Batch #4 Cookie manipulation: estimated time < 10 minutes (47 tests, 12 inputs)

Batch #4 Cookie manipulation: 47 vulnsigs tests, completed 568 requests, 96 seconds. Completed 568 requests of 466 estimated requests (121.888%). XSS optimization removed 58 links. All tests completed.

Batch #4 Header manipulation: estimated time < 10 minutes (47 tests, 5 inputs)

Batch #4 Header manipulation: 47 vulnsigs tests, completed 847 requests, 144 seconds. Completed 847 requests of 650 estimated requests (130.308%). XSS optimization removed 290 links. All tests completed.

Batch #4 shell shock detector: estimated time < 1 minute (1 tests, 5 inputs)

Batch #4 shell shock detector: 1 vulnsigs tests, completed 7 requests, 15 seconds. Completed 7 requests of 5 estimated requests (140%). All tests completed.

Batch #4 shell shock detector(form): estimated time < 1 minute (1 tests, 0 inputs)

Batch #4 shell shock detector(form): 1 vulnsigs tests, completed 0 requests, 0 seconds. No tests to execute.

Login Brute Force manipulation estimated time: no tests enabled

Login Brute Force manipulation estimated time: no tests enabled

Batch #4 insecurely served cred forms detector (no auth): estimated time < 1 minute (1 tests, 2 inputs)

Batch #4 insecurely served cred forms detector (no auth): 1 vulnsigs tests, completed 2 requests, 0 seconds. Completed 2 requests of 2 estimated requests (100%). All tests completed.

Cookies Without Consent no tests enabled.

Batch #5 HTTP Time Bandit: estimated time < 1 minute (1 tests, 10 inputs)

Batch #5 HTTP Time Bandit: 1 vulnsigs tests, completed 200 requests, 9 seconds. No tests to execute.

Path manipulation: Estimated requests (payloads x links): files with extension:(0 x 4) + files:(0 x 8) + directories:(4 x 17) + paths:(11 x 25) = total (343)

Batch #5 Path XSS manipulation: estimated time < 1 minute (15 tests, 25 inputs)

Batch #5 Path XSS manipulation: 15 vulnsigs tests, completed 342 requests, 19 seconds. Completed 342 requests of 343 estimated requests (99.7085%). All tests completed.

Path manipulation: Estimated requests (payloads x links): files with extension:(0 x 4) + files:(0 x 8) + directories:(1 x 17) + paths:(0 x 25) = total (17)

Batch #5 Tomcat Vuln manipulation: estimated time < 1 minute (1 tests, 25 inputs)

Batch #5 Tomcat Vuln manipulation: 1 vulnsigs tests, completed 17 requests, 1 seconds. Completed 17 requests of 17 estimated requests (100%). All tests completed.

Path manipulation: Estimated requests (payloads x links): files with extension:(0 x 4) + files:(0 x 8) + directories:(16 x 17) + paths:(0 x 25) = total (272)

Batch #5 Time based path manipulation: estimated time < 1 minute (16 tests, 14 inputs)

Batch #5 Time based path manipulation: 16 vulnsigs tests, completed 48 requests, 100 seconds. Completed 48 requests of 272 estimated requests (17.6471%). All tests completed.

Path manipulation: Estimated requests (payloads x links): files with extension:(1 x 4) + files:(8 x 8) + directories:(65 x 17) + paths:(11 x 25) = total (1448)

Batch #5 Path manipulation: estimated time < 10 minutes (85 tests, 25 inputs)

Batch #5 Path manipulation: 85 vulnsigs tests, completed 751 requests, 49 seconds. Completed 751 requests of 1448 estimated requests (51.8646%). All tests completed.

Batch #5 WebCgiHrs: estimated time < 1 minute (1 tests, 1 inputs)

Batch #5 WebCgiHrs: 1 vulnsigs tests, completed 3 requests, 1 seconds. Completed 3 requests of 50 estimated requests (6%). All tests completed.

Batch #5 WebCgiGeneric: estimated time < 2 hours (1119 tests, 1 inputs)

Batch #5 WebCgiGeneric: 1119 vulnsigs tests, completed 7223 requests, 595 seconds. Completed 7223 requests of 43425 estimated requests (16.6333%). All tests completed.

Batch #5 Open Redirect analysis: estimated time < 1 minute (2 tests, 2 inputs)

Batch #5 Open Redirect analysis: 2 vulnsigs tests, completed 0 requests, 5 seconds. Completed 0 requests of 4 estimated requests (0%). All tests completed.

Duration of Crawl Time: 2048.00 (seconds)

Duration of Test Phase: 3209.00 (seconds)

Total Scan Time: 5257.00 (seconds)

Total requests made: 12373

Average server response time: 0.50 seconds

Average browser load time: 0.64 seconds

INFO 150028 Cookies Collected (1)

INFO 150028 Cookies Collected

Finding #	15242333(1004327969)	Severity	Information Gathered - Level 1
Unique #	ba7ed23b-6e05-4705-805d-003586ea8762		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The cookies listed in the Results section were set by the web application during the crawl phase.

Impact

Cookies may potentially contain sensitive information about the user.

Note: Long scan duration can occur if a web application sets a large number of cookies (e.g., 25 cookies or more) and QIDs 150002, 150046, 150047, and 150048 are enabled.

Solution

Review cookie values to ensure they do not include sensitive information. If scan duration is excessive due to a large number of cookies, consider excluding QIDs 150002, 150046, 150047, and 150048.

Results

Total cookies: 2
PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; path=/ First set at URL: https://10.1.242.98/
Path=/; secure; HttpOnly First set at URL: https://10.1.242.98/

INFO 150115 Authentication Form found (1)

INFO 150115 Authentication Form found

Finding #	15242338(1004328175)	Severity	Information Gathered - Level 1
Unique #	670709a1-3369-41ae-966b-cea2050dc836		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

Authentication Form was found during the web application crawling.

Impact

N/A

Solution

N/A

Results

Authentication form found at: https://10.1.242.98/index.php/app/login
Action uri: https://10.1.242.98/index.php/app/login
Fields: LoginForm[username], LoginForm[password], yt0

INFO 150148 AJAX Links Crawled (1)

INFO 150148 AJAX Links Crawled

Finding #	15242329(1004327965)	Severity	Information Gathered - Level 1
Unique #	e3bb0d32-5f57-4288-a0bb-67fc6918183d		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The list of unique AJAX links crawled by the scanner appears in the Results section. The link may be either a URL with fragment (#) or a Selenium script. To open a URL with fragment, open it in browser. To open a Selenium script, use Qualys Browser Recorder Chrome extension. The number of AJAX links reported is limited to 1000.

Impact

N/A

Solution

N/A

Results

Number of ajax links: 2

```
<?xml version="1.0" encoding="UTF-8"?><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"><head profile="http://selenium-ide.openqa.org/profiles/test-case"><title>replay</title></head><body><table><thead><tr><td colspan="3">AJAX Link</td></tr></thead><tbody><tr><td>open</td><td>https://10.1.242.98/index.php/app/login</td><td></td></tr><tr><td>pause</td><td>1000</td><td></td></tr><tr><td>click</td><td>xpath=//BODY[1]/DIV[1]/DIV[1]/DIV[1]/BUTTON[1]</td><td></td></tr></tbody></table></body></html>
```

<?xml version="1.0" encoding="UTF-8"?><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"><head profile="http://selenium-ide.openqa.org/profiles/test-case"><title>replay</title></head><body><table><thead><tr><td colspan="3">AJAX Link</td></tr></thead><tbody><tr><td>open</td><td>https://10.1.242.98/index.php/app/login</td><td></td></tr><tr><td>pause</td><td>1000</td><td></td></tr><tr><td>click</td><td>xpath=//BODY[1]/DIV[1]/DIV[1]/DIV[1]/BUTTON[2]</td><td></td></tr></tbody></table></body></html>

Number of ajax links discarded due to crawl optimization: 0

Smart Scan Optimizations - All Optimizations enabled.

INFO 150152 Forms Crawled (1)

INFO 150152 Forms Crawled

Finding #	15242332(1004327968)	Severity	Information Gathered - Level 1
Unique #	d58a04ea-558c-4736-8b2e-870c26de1a29		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The Results section lists the unique forms that were identified and submitted by the scanner. The forms listed in this QID do not include authentication forms (i.e. login forms), which are reported separately under QID 150115.

The scanner does a redundancy check on forms by inspecting the form fields. Forms determined to be the redundant based on identical form fields will not be tested. If desired, you can enable 'Include form action URI in form uniqueness calculation' in the WAS option profile to have the scanner also consider the form's action attribute in the redundancy check.

NOTE: Any regular expression specified under 'Redundant Links' are not applied to forms. Forms (unique or redundant) are not reported under QID 150140.

Impact

N/A

Solution

N/A

Results

Total internal forms seen (this count includes duplicate forms): 0

Crawled forms (Total: 0)

NOTE: This does not include authentication forms. Authentication forms are reported separately in QID 150115

INFO 150172 Requests Crawled (1)

WAS Scan Report

INFO

150172 Requests Crawled

Finding #	15242336(1004327973)	Severity	Information Gathered - Level 1
Unique #	8f219036-8bd6-4116-b254-93703eae330b		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The QID reports list of requests crawled by the Web application scanner appear in the Results section.

Impact

N/A

Solution

N/A

Results

Number of crawled XHRs (XHRs, Fetch and External XHRs): 1

Fetch Requests: 1

Method GET URI https://10.1.242.98/api/setCookie (Count: 1)

INFO

150176 In-scope JavaScript Libraries Detected (1)

INFO 150176 In-scope JavaScript Libraries Detected

Finding #	15242339(1004328176)	Severity	Information Gathered - Level 1
Unique #	5710766d-a364-4700-a1b6-a352e00a5b8d		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-200		
OWASP	-		
WASC	-		

Details

Threat

WAS will report "in-scope" JavaScript libraries discovered by the scanner during crawling and are provided in the Results section. In-scope means, links that are considered to be "in-scope" per the configuration set up for the Web Application. The discovered libraries are reported only once, based on the page on which they were first detected.

Each library is reported along with other information such as the URL of page on which it was first found, the version, and the URL of the .js file.

Impact

When including third-party JavaScript libraries, the application must effectively trust those libraries added. Without sufficient protection mechanisms, the functionality may be malicious in nature (i.e. either by coming from an untrusted source, being spoofed, or being modified in transit from a trusted source).

Solution

Use digital signatures or similar mechanisms to verify the software or data is from the expected source and has not been altered. Ensure libraries and dependencies, are consuming trusted repositories. If you have a higher risk profile, consider hosting an internal known-good repository that's vetted.

Results

Number of unique JS libraries: 2
Javascript library : Bootstrap
Version : 5.1.3
Script uri : https://10.1.242.98/resources/4ef8da87/js/bootstrap.min.js
Found on the following page(only first page is reported):
https://10.1.242.98/index.php/app/login

=====

Javascript library : jQuery
Version : 3.5.1
Script uri : https://10.1.242.98/resources/4ef8da87/js/jquery-3.5.1.min.js
Found on the following page(only first page is reported):
https://10.1.242.98/index.php/app/login

=====

INFO 150247 Web Server and Technologies Detected (1)

INFO 150247 Web Server and Technologies Detected

Finding #	15242340(1004328177)	Severity	Information Gathered - Level 1
Unique #	4f1191fd-8995-4eb6-bf2f-abe6c861fc16		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-200		
OWASP	-		
WASC	-		

Details

Threat

Information disclosure is an application weakness in revealing sensitive data, such as technical details of the system or environment.

This check reports the various technologies used by the web application based on the information available in different components of the Request-Response.

Impact

An attacker may use sensitive data to exploit the target web application, its hosting network, or its users.

Solution

Ensure that your web servers do not reveal any sensitive information about your technology stack and system details

Please review the issues reported below:

Results

Number of technologies detected: 3
Technology name: Bootstrap
Matched Components:
script tag match:
<script type="text/javascript" src="/resources/4ef8da87/js/bootstrap-table.js"></script>
<script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.bundle.js"></script>
<script type="text/javascript" src="/resources/4ef8da87/js/bootstrap.min.js"></script>
Matched links: reporting only first 3 links
https://10.1.242.98/index.php/app/login

Technology name: OpenResty
Technology version: OpenResty 1.15.8.2
Matched Components:
header match:
Server:openresty/1.15.8.2
Matched links: reporting only first 3 links
https://10.1.242.98/
https://10.1.242.98/index.php/app/login
https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js

Technology name: PHP
Technology version: PHP 8.1.32
Matched Components:
header match:
X-Powered-By:PHP/8.1.32
cookie match:
PHPSESSID:3100sg2cit57j55sj5ukek1v9h
Matched links: reporting only first 3 links
https://10.1.242.98/
https://10.1.242.98/index.php/app/login
https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js

INFO 150528 Server Returns HTTP 4XX Error Code During Scanning (1)

INFO 150528 Server Returns HTTP 4XX Error Code During Scanning

Finding #	15713256(1004327962)	Severity	Information Gathered - Level 1
Unique #	d30aa0bb-a310-4eaf-beb2-b5c5c0037b50		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

During the WAS scan, links with HTTP 4xx response code were observed and these are listed in the Results section. The HTTP 4xx message indicates a client error. The list of supported 4xx response code are as below:

- 400 - Bad Request
- 401 - Unauthorized
- 403 - Forbidden
- 404 - Not Found
- 405 - Method Not Allowed
- 407 - Proxy Authentication Required
- 408 - Request Timeout
- 413 - Payload Too Large
- 414 - URI Too Long

Impact

The presence of a HTTP 4xx error during the crawl phase indicates that some problem exists on the website that will be encountered during normal usage of the Web application. Note WAS depends on responses to detect many vulnerabilities if the link does not respond with an expected response then any vulnerabilities present on such links may not be detected.

Solution

Review each link to determine why the client encountered an error while requesting the link. Additionally review and investigate the results of QID 150042 which lists 5xx errors, QID 150019 which lists unexpected response codes and QID 150097 which lists a potential blocked request.

Results

Number of links with 4xx response code: 1
(Only first 50 such links are listed)

404 https://10.1.242.98/api/setCookie

INFO 150546 First Link Crawled Response Code Information (1)

INFO 150546 First Link Crawled Response Code Information

Finding #	15242334(1004327971)	Severity	Information Gathered - Level 1
Unique #	42d8561a-dda2-484c-a94c-0251cfb5660b		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The Web server returned the following information from where the Web application scanning engine initiated. Information reported includes First Link Crawled, response Code, response Header, and response Body (first 500 characters). The first link crawled is the "Web Application URL (or Swagger file URL)" set in the Web Application profile.

Impact

An erroneous response might be indicative of a problem in the Web server, or the scan configuration.

Solution

Review the information to check if this is in line with the expected scan configuration. Refer to the output of QIDs 150009, 150019, 150021, 150042 and 150528 (if present) for additional details.

Results

Base URI: https://10.1.242.98/
Response Code: 302
Response Header:
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:14:14 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: https://10.1.242.98/index.php/app/login
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: frame-ancestors 'none';
Set-Cookie: Path=/; HttpOnly; Secure
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=/
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/

Response Body:

INFO 150621 List of JavaScript Links (1)

INFO 150621 List of JavaScript Links

Finding #	15242337(1004327974)	Severity	Information Gathered - Level 1
Unique #	c1905b17-e6f4-40d9-b1f6-41cee8d6786e		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

This QID reports all the JavaScript links that are in-scope of this scan.

Impact

JavaScript links may pose security risks such as XSS, CSRF.

Solution

Verify JavaScript links are intentional and required for your web application.
Review any third party scripts that are hosted on your local server instead of using CDN.
Update all the JavaScript libraries with latest version as applicable.

Results

JavaScript Links were found while crawling.
Total Number of Links: 6
https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js
https://10.1.242.98/resources/4ef8da87/js/bootstrap.bundle.js
https://10.1.242.98/resources/30d9a0ac/js/login.js
https://10.1.242.98/resources/4ef8da87/js/jquery-3.5.1.min.js
https://10.1.242.98/resources/4ef8da87/js/bootstrap.min.js
https://10.1.242.98/resources/30d9a0ac/js/error.js

INFO 150845 Business logic abuse potential due to presence of external domains detected (1)

INFO 150845 Business logic abuse potential due to presence of external domains detected

Finding #	15713259(1004328189)	Severity	Information Gathered - Level 1
Unique #	214468e3-c250-4384-93dc-6814d8e449c3		
Group	Scan Diagnostics	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	A4 Insecure Design		
WASC	-		

Details

Threat

External domains detected in the application. Using external domains in an application introduces risk by potentially exposing the application to external threats and dependencies, which can be exploited for malicious purposes such as data exfiltration, phishing, or compromise of application integrity. These vulnerabilities arise from inadequate validation, reliance on unsecured external services, and the application's failure to enforce strict security controls over external interactions.

Impact

N/A

Solution

Audit external domains accessed by your application. If possible launch scans against those.

Results

External domains could be involved in potential business logic abuse.
cdnjs.cloudflare.com
fonts.googleapis.com
fonts.gstatic.com
rcisuat.taiwanlife.com

Security Weaknesses (9)

INFO 150210 Information Disclosure via Response Header (1)

INFO 150210 Information Disclosure via Response Header

Finding #	15242325(1004327960)	Severity	Information Gathered - Level 3
Unique #	231b1e89-fe29-49de-bda3-0a708bad730d		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-16, CWE-201		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		

Details

Threat

HTTP response headers like 'Server', 'X-Powered-By', 'X-AspNetVersion', 'X-AspNetMvcVersion' could disclose information about the platform and technologies used by the website. The HTTP response include one or more such headers.

Impact

The headers can potentially be used by attackers for fingerprinting and launching attacks specific to the technologies and versions used by the web application. These response headers are not necessary for production sites and should be disabled.

Solution

Disable such response headers, remove them from the response, or make sure that the header value does not contain information which could be used to fingerprint the server-side components of the web application.

Results

One or more response headers disclosing information about the application platform were present on the following pages:
(Only first 50 such pages are reported)

GET https://10.1.242.98/ response code: 302
Server: openresty/1.15.8.2
X-Powered-By: PHP/8.1.32

GET https://10.1.242.98/index.php/app/login response code: 200
Server: openresty/1.15.8.2
X-Powered-By: PHP/8.1.32

GET https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.bundle.js response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/css/style.css response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/30d9a0ac/js/login.js response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/js/jquery-3.5.1.min.js response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/30d9a0ac/images/icon/favicon.png response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.min.js response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-icons.css response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-table.css response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/app/index response code: 302
Server: openresty/1.15.8.2
X-Powered-By: PHP/8.1.32

GET https://10.1.242.98/protected/views/app/images/background02.png response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/font/Noto_Sans_TC/NotoSansTC-Regular.otf response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/api/setCookie response code: 404
Server: openresty/1.15.8.2
X-Powered-By: PHP/8.1.32

GET https://10.1.242.98/resources/30d9a0ac/js/error.js response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/resources/4ef8da87/images/errorBac.png response code: 200
Server: openresty/1.15.8.2

GET https://10.1.242.98/app/login response code: 200
Server: openresty/1.15.8.2
X-Powered-By: PHP/8.1.32

INFO 150261 Subresource Integrity (SRI) Not Implemented (1)

INFO 150261 Subresource Integrity (SRI) Not Implemented

Finding #	15242346(1004328184)	Severity	Information Gathered - Level 3
Unique #	f352a468-d4e0-435d-ba4e-45719ed4a749		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-693		
OWASP	A1 Broken Access Control		
WASC	-		

Details

Threat

The integrity attribute is missing in script and/or link elements. Subresource Integrity (SRI) is a standard browser security feature that verifies the value of the integrity attribute in

Impact

Absence of SRI checks means it is impossible to verify that the third-party resources are delivered without any unexpected manipulation.

Solution

All script and link elements that load external content should include the integrity attribute to ensure that the content is trustworthy.

More information:
[Subresource Integrity article by Mozilla](#)
[OWASP Third-Party JavaScript Management Cheat Sheet](#)

Results

Externally loaded Javascript and CSS resources without integrity checks:

Parent link : https://10.1.242.98/index.php/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

Parent link : https://10.1.242.98/index.php/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

Parent link : https://10.1.242.98/index.php/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

Parent link : https://10.1.242.98/index.php/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

Parent link : https://10.1.242.98/index.php/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

Parent link : https://10.1.242.98/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

Parent link : https://10.1.242.98/index.php/app/login
Found following resource links without integrity checks (only first 10 links are reported)
https://cdnjs.cloudflare.com/ajax/libs/autosize.js/4.0.2/autosize.min.js

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

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

Finding #	15242348(1004328186)	Severity	Information Gathered - Level 2
Unique #	ed2da081-d80d-42d8-b22c-e1682e88a7ff		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-16, CWE-1032		
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://10.1.242.98/index.php/app/login response code: 200
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:14:17 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: frame-ancestors 'none';
Set-Cookie: Path=/; HttpOnly; Secure
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/index.php/app/
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=
Content-Encoding: gzip

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

GET https://10.1.242.98/index.php/app/login response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.bundle.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/style.css response code: 200
GET https://10.1.242.98/resources/30d9a0ac/js/login.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/jquery-3.5.1.min.js response code: 200
GET https://10.1.242.98/resources/30d9a0ac/images/icon/favicon.png response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.min.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-icons.css response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-table.css response code: 200
GET https://10.1.242.98/protected/views/app/images/background02.png response code: 200
GET https://10.1.242.98/resources/4ef8da87/font/Noto_Sans_TC/NotoSansTC-Regular.otf response code: 200
GET https://10.1.242.98/api/setCookie response code: 404
GET https://10.1.242.98/resources/30d9a0ac/js/error.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/images/errorBac.png response code: 200
GET https://10.1.242.98/app/login response code: 200

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

INFO 150206 Content-Security-Policy Not Implemented

Finding #	15713258(1004328188)	Severity	Information Gathered - Level 2
Unique #	af65722c-440f-45b3-a0ee-d5f4829c0f14		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-16, CWE-1032		
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://developers.google.com/web/fundamentals/security/csp/>

Results

Content-Security-Policy: Header missing
Response headers on link: GET https://10.1.242.98/api/setCookie response code: 404
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:16:22 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Encoding: gzip
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=/
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/

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

GET https://10.1.242.98/api/setCookie response code: 404

INFO 150208 Missing header: Referrer-Policy (1)

INFO 150208 Missing header: Referrer-Policy

Finding #	15242326(1004327961)	Severity	Information Gathered - Level 2
Unique #	2ede7d4a-d742-4c5e-9f04-d5cae70a4066		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-16, CWE-1032		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		

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://10.1.242.98/index.php/app/login response code: 200
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:14:17 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: frame-ancestors 'none';
Set-Cookie: Path=/; HttpOnly; Secure
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/index.php/app/
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=/
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/
Content-Encoding: gzip

Header missing on the following link(s):

(Only first 50 such pages are listed)

GET https://10.1.242.98/index.php/app/login response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.bundle.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/style.css response code: 200
GET https://10.1.242.98/resources/30d9a0ac/js/login.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/jquery-3.5.1.min.js response code: 200
GET https://10.1.242.98/resources/30d9a0ac/images/icon/favicon.png response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.min.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-icons.css response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-table.css response code: 200
GET https://10.1.242.98/protected/views/app/images/background02.png response code: 200
GET https://10.1.242.98/resources/4ef8da87/font/Noto_Sans_TC/NotoSansTC-Regular.otf response code: 200
GET https://10.1.242.98/api/setCookie response code: 404
GET https://10.1.242.98/resources/30d9a0ac/js/error.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/images/errorBac.png response code: 200
GET https://10.1.242.98/app/login response code: 200

INFO 150248 Missing header: Permissions-Policy (1)

WAS Scan Report

INFO

150248 Missing header: Permissions-Policy

Finding #	15242343(1004328180)	Severity	Information Gathered - Level 2
Unique #	138326e5-733e-4fe8-ac0b-0d29b9da8a9e		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-284		
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://10.1.242.98/index.php/app/login response code: 200
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:14:17 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: frame-ancestors 'none';
Set-Cookie: Path=/; HttpOnly; Secure
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/index.php/app/
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=
Content-Encoding: gzip

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

GET https://10.1.242.98/index.php/app/login response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap-table.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.bundle.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/style.css response code: 200
GET https://10.1.242.98/resources/30d9a0ac/js/login.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/jquery-3.5.1.min.js response code: 200
GET https://10.1.242.98/resources/30d9a0ac/images/icon/favicon.png response code: 200
GET https://10.1.242.98/resources/4ef8da87/js/bootstrap.min.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-icons.css response code: 200
GET https://10.1.242.98/resources/4ef8da87/css/bootstrap-table.css response code: 200
GET https://10.1.242.98/protected/views/app/images/background02.png response code: 200
GET https://10.1.242.98/resources/4ef8da87/font/Noto_Sans_TC/NotoSansTC-Regular.otf response code: 200

WAS Scan Report

GET https://10.1.242.98/api/setCookie response code: 404
GET https://10.1.242.98/resources/30d9a0ac/js/error.js response code: 200
GET https://10.1.242.98/resources/4ef8da87/images/errorBac.png response code: 200
GET https://10.1.242.98/app/login response code: 200

INFO 150126 Links With High Resource Consumption (1)

INFO 150126 Links With High Resource Consumption

Finding #	16190222(1004328181)	Severity	Information Gathered - Level 1
Unique #	4e95cecb-ae02-4178-a435-1ace9db2dff2		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	-		
OWASP	-		
WASC	-		

Details

Threat

The list of links with lowest bytes/sec which are assumed to be resources with highest resource consumption. The links in the list have slower transfer times speeds to an average resource on the server. This may indicate that the links are more CPU or DB intensive than majority of links.

The latency of the network and file size have no effect on calculations.

Impact

The links with high resource consumption could be used to perform DOS on the server by just performing GET Flooding. Attackers could more easily take the server down if there are huge resource hogs on it, performing less request.

Solution

Find the root cause of resources slow download speed.

If the cause is a real CPU strain or complex DB queries performed, there may be a need for re-engineering of the web application or defense measures should be in place. Examples of defense against DOS that is targeted towards high resource consumption links are Load Balancers and Rate Limiters.

Results

10141.700000 bytes/sec https://10.1.242.98/api/setCookie
33657.000000 bytes/sec https://10.1.242.98/app/login
33686.400000 bytes/sec https://10.1.242.98/index.php/app/login
172849.100000 bytes/sec https://10.1.242.98/resources/30d9a0ac/js/error.js
396289.800000 bytes/sec https://10.1.242.98/resources/30d9a0ac/js/login.js
18126398.200000 bytes/sec https://10.1.242.98/resources/30d9a0ac/images/icon/favicon.png
18576965.700000 bytes/sec https://10.1.242.98/resources/4ef8da87/js/bootstrap.bundle.js
30718649.800000 bytes/sec https://10.1.242.98/resources/4ef8da87/images/errorBac.png
51156222.600000 bytes/sec https://10.1.242.98/protected/views/app/images/background02.png
167688258.500000 bytes/sec https://10.1.242.98/resources/4ef8da87/font/Noto_Sans_TC/NotoSansTC-Regular.otf

INFO 150245 Missing header: X-Frame-Options (1)

INFO 150245 Missing header: X-Frame-Options

Finding #	15242342(1004328179)	Severity	Information Gathered - Level 1
Unique #	b8b74c3a-c6ad-4018-82e8-de7efd1c67cb		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-693		
OWASP	A5 Security Misconfiguration		
WASC	WASC-15 APPLICATION MISCONFIGURATION		

Details

Threat

The X-Frame-Options header is not set in the HTTP response, meaning the page can potentially be loaded into an attacker-controlled frame. This could lead to clickjacking, where an attacker adds an invisible layer on top of the legitimate page to trick users into clicking on a malicious link or taking a harmful action.

Note: Only responses with status code 200 ok are tested and reported for 150245 and 150124

Impact

Without an X-Frame-Options response header, clickjacking may be possible. However, if the application properly uses the Content-Security-Policy "frame-ancestors" directive, then modern web browsers would stop the page from being framed and prevent clickjacking.

Solution

The X-Frame-Options allows three values: DENY, SAMEORIGIN and ALLOW-FROM. It is recommended to use DENY, which prevents all domains from framing the page or SAMEORIGIN, which allows framing only by the same site. DENY and SAMEORIGIN are supported by all browsers. Using ALLOW-FROM is not recommended because not all browsers support it.

Note: To avoid a common X-Frame-Options implementation mistake, see <https://blog.qualys.com/securitylabs/2015/10/20/clickjacking-a-common-implementation-mistake-that-can-put-your-websites-in-danger>.

Results

X-Frame-Options header is missing or not set to DENY or SAMEORIGIN for the following pages:
(Only first 10 such pages are reported)

GET https://10.1.242.98/app/login
Response code: 200
Response headers:
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:18:51 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: frame-ancestors 'none';
Set-Cookie: Path=/; HttpOnly; Secure
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/app/
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=/
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/
Content-Encoding: gzip

GET https://10.1.242.98/index.php/app/login
Response code: 200
Response headers:
Server: openresty/1.15.8.2
Date: Tue, 25 Mar 2025 09:14:17 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/8.1.32
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache

WAS Scan Report

Strict-Transport-Security: max-age=31536000; includeSubDomains
Content-Security-Policy: frame-ancestors 'none';
Set-Cookie: Path=/; HttpOnly; Secure
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/index.php/app/
Set-Cookie: PHPSESSID=3100sg2cit57j55sj5ukek1v9h; secure; HttpOnly; domain=10.1.242.98; path=/
Set-Cookie: Path=/; secure; HttpOnly; domain=10.1.242.98; path=/
Content-Encoding: gzip

INFO

150277 Cookie without SameSite attribute (1)

INFO

150277 Cookie without SameSite attribute

Finding #	15242331(1004327967)	Severity	Information Gathered - Level 1
Unique #	dbfd1c3a-80d6-4801-84c9-c796735f5397		
Group	Security Weaknesses	Detection Date	25 Mar 2025 17:11 GMT+0800
CWE	CWE-16, CWE-1032		
OWASP	A5 Security Misconfiguration		
WASC	-		

Details

Threat

The cookies listed in the Results section are missing the SameSite attribute.

Impact

The SameSite cookie attribute is an effective countermeasure against cross-site request forgery (CSRF) attacks. Note that a missing SameSite attribute does not mean the web application is automatically vulnerable to CSRF. The scanner will report QID 150071 if a CSRF vulnerability is detected.

Solution

Consider adding the SameSite attribute to the cookie(s) listed.

More information:
[DZone article](#)
[OWASP CSRF Prevention Cheat Sheet](#)

Results

Total cookies: 2
PHPSESSID=3100sg2cit57j55sj5ukek1v9h; path=/; domain=10.1.242.98; secure; httponly | First set at URL: https://10.1.242.98/
Path=/; path=/; domain=10.1.242.98; secure; httponly | First set at URL: https://10.1.242.98/

Appendix

Scan Details

RCIS_Web Application Vulnerability Scan - Mar 25, 2025	
Reference	was/1742893665480.70402447
Date	25 Mar 2025 17:11 GMT+0800
Mode	On-Demand
Progressive Scanning	Disabled
Type	Vulnerability
Authentication	None
Scanner Appliance	HQ0LUX246 (IP: 10.1.119.2, Scanner: 14.6.12-1, WAS: 9.11.34-1, Signatures: 2.6.283-2)
Profile	twlife
DNS Override	-
Duration	01:27:40
Status	Finished
Authentication Status	None

Option Profile Details

Form Submission	BOTH
Form Crawl Scope	Do not include form action URI in uniqueness calculation
Maximum links to test in scope	600
User Agent	-
Request Parameter Set	Initial Parameters
Document Type	Ignore common binary files
Enhanced Crawling	Disabled
SmartScan	Enabled
SmartScan Depth	10
Timeout Error Threshold	300
Unexpected Error Threshold	600
Performance Settings	Pre-defined
Scan Intensity	High
Bruteforce Option	Disabled
Detection Scope	Custom Search Lists
Include additional XSS payloads	No
Inclusion Search List Names	-
Exclusion Search List Names	SSL certificates verify, twlife : exclude list
Inclusion Search List QIDs	-
Exclusion Search List QIDs	38167, 38169, 38170, 38171, 38172, 38716, 38173, 38685, 38174, 151040, 150263, 150004, 150476
Credit Card Numbers Search	Off
Social Security Numbers (US) Search	Off

Web Application Details: Web地址正規化後台(RCIS_DEV)

Name	Web地址正規化後台(RCIS_DEV)
ID	1288134970

WAS Scan Report

URL	https://10.1.242.98/
Owner	oliver kuo (ctbcf_ik)
Scope	Limit to URL hostname
Tags	Taiwanlife, dev
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.
	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.

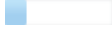




WAS Scan Report

	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.