



WEBSEC

WEB APPLICATION REMEDIATION VERIFICATION REPORT

Client: CareValidate, Inc.

Date: June 30, 2023

Contact: rsalgado@websec.ca

C O N F I D E N T I A L

Report Status

Version:	1.3
Status:	Final
Date:	June-30-23
Contact:	Roberto Salgado (rsalgado@websec.ca)

Version	Date	Description of the Task	Author	Pages
0.1	06/15/2023	Created document	Wyatt Harvey	43
1.0	06/19/2023	Reviewed document	Sebastian Bethell	43
1.1	06/23/2023	Updated document	Wyatt Harvey	40
1.2	06/26/2023	Reviewed document	Eric Vuong	36
1.3	06/30/2023	Revised document	Sebastian Bethell	36

PURPOSE OF THE DOCUMENT

This document presents the results of a Remediation Verification Test as requested by Samantha Whitmore (samantha@carevalidate.com) from CareValidate, Inc.

DISCLAIMER

This document, produced by Websec Information Security Services, Inc. ("Websec"), is copyrighted and confidential. It has been prepared solely for the internal use of CareValidate, Inc. ("CareValidate") in accordance with its specific purpose. Unauthorized distribution, reproduction, or modification of this document is strictly prohibited.

The findings in this report are based on the state of the systems at the time of testing and do not provide a guarantee of ongoing security. This report should not be relied upon as an exhaustive statement of risks or vulnerabilities. Although Websec takes precautions when preparing this document, Websec assumes no responsibility for any omission(s), error(s), or the impact of the recommendation(s). CareValidate is solely responsible for determining what changes or

improvements (if any) they should implement in light of their objectives and Websec's recommendations.

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY	3
1.1. SUMMARY OF FINDINGS	3
1.2. HIGH-LEVEL RECOMMENDATIONS	6
2. ENGAGEMENT SCOPE.....	7
3. VULNERABILITY SUMMARY	8
4. FINDINGS BY HOST.....	9
4.1. US-CENTRAL1-CONTACT-TRACING-PROD.CLOUDFUNCTIONS.NET (216.239.36.54)	9
4.1.1. <i>Insufficient Anti-Automation Controls (CVSS: 5.3)</i>	10
4.1.2. <i>Usage of Weak Cipher Suites (CVSS: 3.7)</i>	15
4.2. CARE360.CAREVALIDATE.COM (18.205.36.100)	19
4.2.1. <i>Usage of Weak Cipher Suites (CVSS: 3.7)</i>	20
4.3. CONTACT-TRACING-PROD.APPSPOT.COM (142.251.45.52)	23
4.3.1. <i>Usage of Weak Cipher Suites (CVSS: 3.7)</i>	24
4.4. API.CARE360.CAREVALIDATE.COM (18.205.222.128).....	28
4.4.1. <i>Usage of Weak Cipher Suites (CVSS: 3.7)</i>	28
5. METHODOLOGY AND STANDARDS	31
5.1. OWASP WEB SECURITY TESTING GUIDE (WSTG)	31
5.2. RISK RATING METHODOLOGY.....	33
5.3. COMMON VULNERABILITY SCORING SYSTEM (CVSS)	34
5.3.1. <i>Vulnerability Severity Ratings</i>	35
5.4. COMMON WEAKNESS ENUMERATION (CWE)	36

1. EXECUTIVE SUMMARY

Websec Information Security Services, Inc. (“Websec”) conducted a Penetration Test (“Test” or “Testing”) on CareValidate, Inc. (“CareValidate”)’s web application from May 10, 2023 to May 11, 2023. Testing was conducted within CareValidate’s production environment and focused on identifying and validating security concerns that could impact CareValidate’s confidentiality, integrity, and availability. Testing was considered a “Black-Box” Test, based on no knowledge of the application or credentials provided by CareValidate.

For this engagement, Websec used the [OWASP Web Security Testing Guide](#) as a baseline for testing; however, Websec considered other security concerns that extend beyond this methodology.

1.1. SUMMARY OF FINDINGS

In the original test, Websec identified four (4) issues of varying severity that affect CareValidate’s security posture - zero (0) critical, zero (0) high, three (3) medium, and one (1) low. Considering all issues found, Websec assessed the overall risk to users and the environment to be medium.

CareValidate remediated the findings, which Websec retested on June 15, 2023. Websec found that CareValidate partially remediated one (1) medium severity issue and fully remediated one (1) medium severity issue. Of the original issues reported, one (1) medium, and one (1) low severity issue remain. One (1) medium severity issue was unreachable during retesting and is considered remediated. Considering all issues found, Websec assessed the overall risk to users and the environment to be low.

The most significant issues include the following:

- The application accepts connections using multiple weak cipher suites. An adversary may be able to intercept and decipher the information exchanged between the application and its users, leading to potential data breaches of unauthorized access to sensitive information.

OBSERVED VULNERABILITIES		
SEVERITY	TOTAL	
Critical	0	0
High	0	0
Medium	1	1
Low	1	4
Informational	0	0
Total	2	5

Figure 1: The severity and number of issues encountered during the assessment.

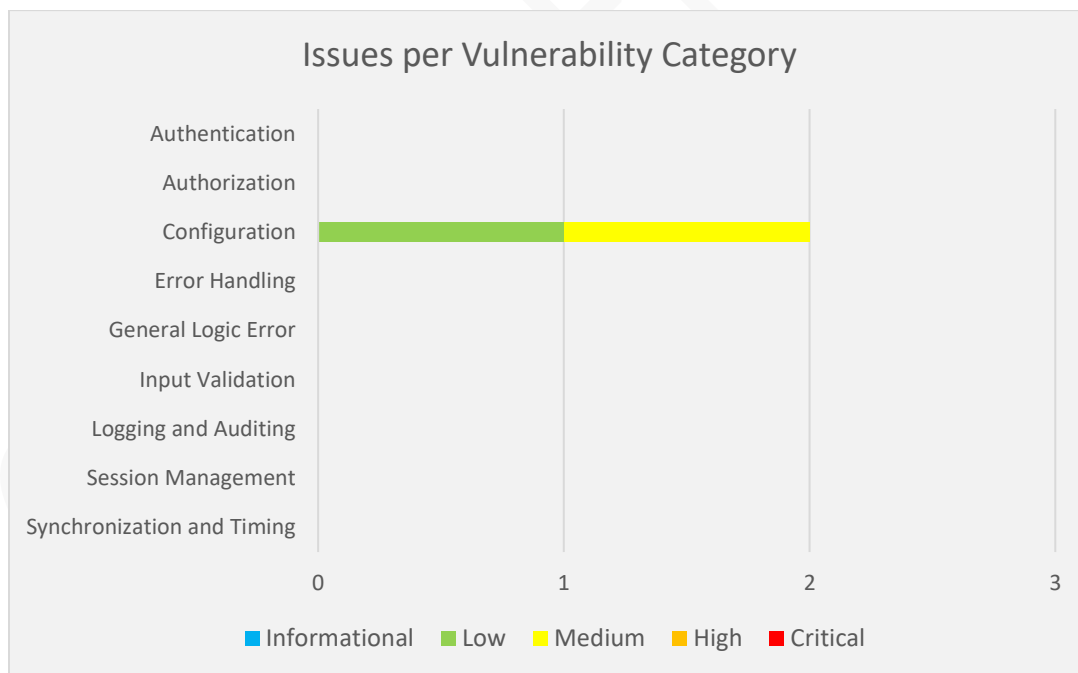


Figure 2: The severity and number of issues encountered by category.

The following Risk Matrix details the recommended action plan for prioritizing each encountered vulnerability. Additional information on how the risk for each vulnerability is calculated can be found in the section, [Risk Rating Methodology](#).

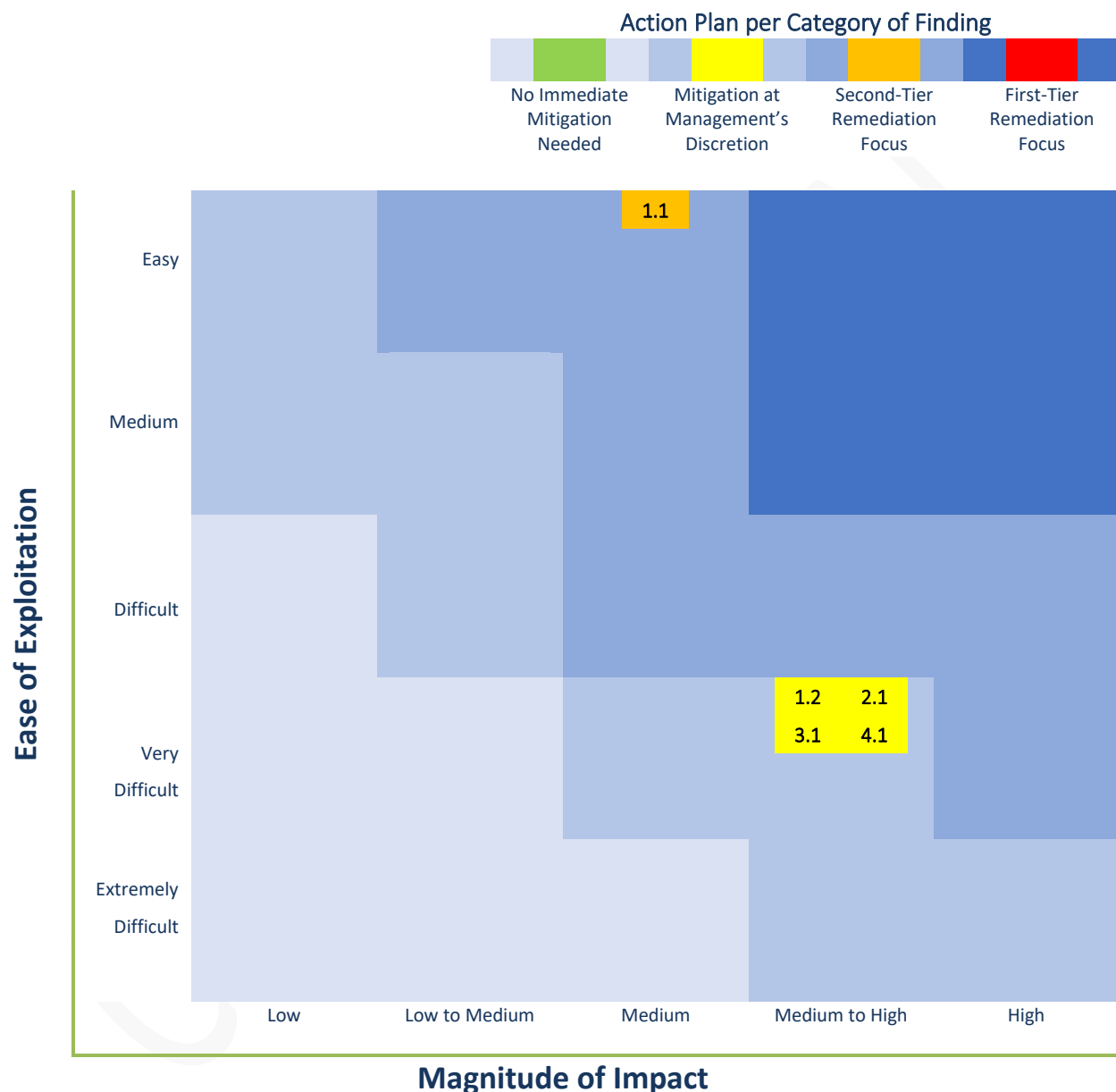


Figure 3: The Impact and Ease assigned. Issues are numbered by their "Findings by Host" subsection.

1.2. HIGH-LEVEL RECOMMENDATIONS

The following recommendations address multiple concerns and would broadly improve CareValidate's security posture:

- Implement a CAPTCHA and/or add a time throttle based on the number of requests sent from a given user to stop automated tools from attempting to enumerate user accounts.
- Ensure that all communications are done over encrypted channels that utilize strong encryption algorithms and protocols.
- Continue regular penetration testing and consider an Application Security Verification Standard assessment and Static Application Security Testing of the codebase.

2. ENGAGEMENT SCOPE

The following assets were determined to be in scope of the engagement:

Host	Description
care360.carevalidate.com	Production Application
us-central1-contact-tracing-prod.cloudfunctions.net	Production Login Portal
contact-tracing-prod.appspot.com	Production GraphQL API
api.care360.carevalidate.com	Production API

CareValidate did not provide Websec with any credentials for testing.

3. VULNERABILITY SUMMARY

Severity Score	Description	Status
5.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)	Insufficient Anti-Automation Controls The application does not employ sufficient anti-automation controls.	Accepted Risk
5.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L)	Unrestricted Google Maps Key Access The application does not restrict the Google Maps API key, which adversaries can use to accumulate large usage costs on the application's behalf.	Remediated
5.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)	GraphQL Introspection is Enabled GraphQL Introspection is enabled on the application and allows a user to view endpoints and data structures that may normally be hidden.	Unreachable During Retesting / Remediated
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Usage of Weak Cipher Suites The server accepts weak cipher suites with known vulnerabilities that affect integrity and confidentiality.	Accepted Risk

4. FINDINGS BY HOST

4.1. US-CENTRAL1-CONTACT-TRACING-PROD.CLOUDFUNCTIONS.NET (216.239.36.54)



us-central1-contact-tracing-prod.cloudfunctions.net (216.239.36.54)

Port(s)	80, 443
Path	/

Severity Score	Description
5.3 (AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)	Insufficient Anti-Automation Controls The application does not employ sufficient anti-automation controls.
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Usage of Weak Cipher Suites The server accepts weak cipher suites with known vulnerabilities that affect integrity and confidentiality.

4.1.1. INSUFFICIENT ANTI-AUTOMATION CONTROLS (CVSS: 5.3)

CVSS Score	Category	Type	Required Access
5.3 <small>(AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N)</small>	Configuration	Improper Control of Interaction Frequency	Network
Risk Evaluation	CWE ID	CAPEC ID	OWASP Top 10
Medium to Low	<u>CWE-779</u>	<u>CAPEC-112</u>	<u>A04:2021-Insecure Design</u>
Location			
<ul style="list-style-type: none"> POST /tracing-getSignUpParameters 			
Description			
<p>Insufficient Anti-Automation occurs when a web application allows an adversary to automate a process that was originally designed to be performed manually. This can allow the adversary to perform actions more frequently than expected.</p>			
Impact			
<p>A persistent adversary may exploit this issue to try to enumerate users of the application.</p>			
Likelihood			
<p>Due to a lack of protections, it is trivial for any visitor to send a large number of requests.</p>			
Remediation			
<ul style="list-style-type: none"> A common practice for protecting against automation attacks is the implementation of CAPTCHA mechanisms. Add a time throttle based on the number of requests sent from a given user. <p>For additional information, please reference:</p> <ul style="list-style-type: none"> https://www.owasp.org/index.php/Brute_force_attack https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks https://owasp.org/www-project-automated-threats-to-web-applications/ 			
Evidence			

Phonebook.cz

[Logout](#)

Phonebook lists all domains, email addresses, or URLs for the given input
You are searching 100 billion records.

Try: [cia.gov](#), [cnn.com](#), [netflix.com](#), [*.ru](#), [*.gov.uk](#), [solarwinds.com](#)

- ☐ Domains
☒ Email Addresses
☐ URLs

info@carevalidate.com
support@carevalidate.com
marybeth@carevalidate.com
jay@carevalidate.com
No more results.

Figure 4: Using OSINT to enumerate email addresses of carevalidate.com.

Please login

Country



United States



Phone Number

+1 919 279 9995



Not authorized.



I'm not a robot



reCAPTCHA
Privacy - Terms

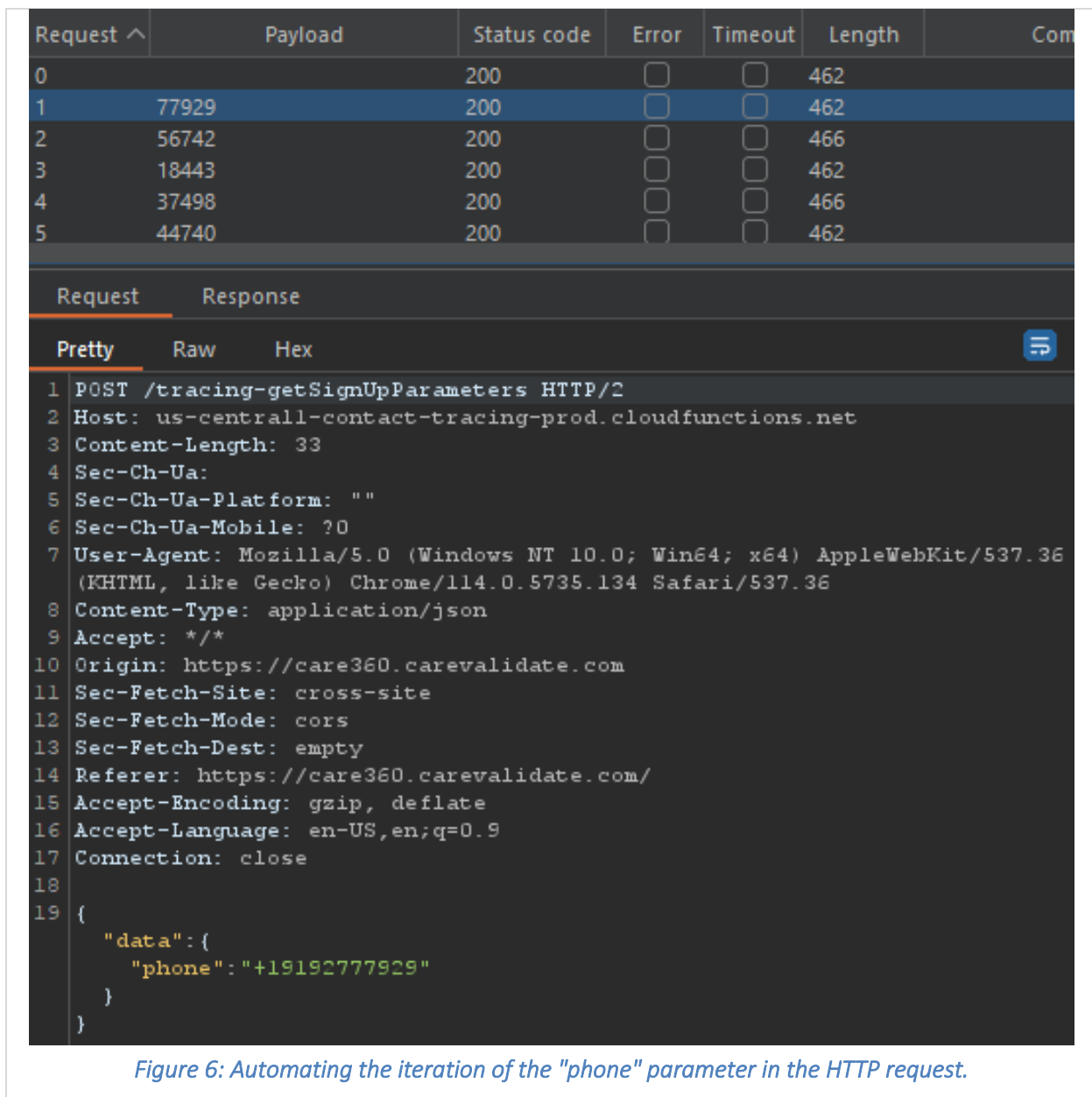
Standard text message rates apply.

Send Text Code

Text Message Code

Login With Code

Figure 5: Entering a phone number and clicking on the "Send text code" button.



Request ^	Payload	Status code	Error	Timeout	Length
495	46219	200	<input type="checkbox"/>	<input type="checkbox"/>	462
496	36678	200	<input type="checkbox"/>	<input type="checkbox"/>	466
497	9433	400	<input type="checkbox"/>	<input type="checkbox"/>	537
498	72198	200	<input type="checkbox"/>	<input type="checkbox"/>	466
499	41526	200	<input type="checkbox"/>	<input type="checkbox"/>	462
500	82184	200	<input type="checkbox"/>	<input type="checkbox"/>	466

Request	Response
Pretty	Hex Render
1	HTTP/2 200 OK
2	Access-Control-Allow-Origin: https://care360.carevalidate.com
3	Vary: Origin
4	Content-Type: application/json; charset=utf-8
5	Function-Execution-Id: as9eh52ovld1
6	X-Cloud-Trace-Context: 29bc7ec38106c863912475c533ec6e2b;o=1
7	Date: Thu, 15 Jun 2023 17:07:22 GMT
8	Server: Google Frontend
9	Cache-Control: private
10	Content-Length: 64
11	Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
12	
13	{
	"result":{
	"success":true,
	"signupParams":{
	"type":"no_account"
	}
	}
	}

Figure 7: Demonstrating that the number of attempts is not blocked, and the captcha is not being validated despite 500 requests.

4.1.2. USAGE OF WEAK CIPHER SUITES (CVSS: 3.7)

CVSS Score	Category	Type	Required Access
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Configuration	Use of a Broken or Risky Cryptographic Algorithm	Network
Risk Evaluation	CWE ID	CAPEC ID	OWASP Top 10
Medium to Low	<u>CWE-327</u>	<u>CAPEC-97</u>	<u>A02:2021-Cryptographic Failures</u>
Location			
Within the following TLS 1.0, TLS 1.1, and TLS 1.2 cipher suites:			
<ul style="list-style-type: none">• TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA• TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA• TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA• TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA• TLS_RSA_WITH_AES_128_GCM_SHA256• TLS_RSA_WITH_AES_256_GCM_SHA384• TLS_RSA_WITH_AES_128_CBC_SHA• TLS_RSA_WITH_AES_256_CBC_SHA• TLS_RSA_WITH_3DES_EDE_CBC_SHA			
Description			
<p>Cryptographic protocols such as SSL and TLS allow for the safe transmission of information between two parties, ensuring that all sensitive information and communication remain unmodified and private. However, the usage of weak ciphers may allow an adversary to read or maliciously alter the data between the two points of communication through a Machine-in-the-Middle (MitM) attack.</p> <p>Cipher suites with any of the following should be avoided:</p>			

- CBC
- DES/3DES
- RC4
- Key lengths under 112 bits

Impact

An adversary may be able to conduct a MitM attack and sniff or modify sensitive information such as login credentials and user emails.

Likelihood

Fully exploiting this vulnerability requires that an adversary be placed between the communication channel between the victim and the server. The victim is actively communicating with the vulnerable server, which significantly increases the difficulty of exploit.

Remediation

- Remove all weak cryptographic ciphers from the list of accepted cipher suites.
- Disable SSL usage, and only utilize cipher suites currently considered secure. These would include all TLS 1.3 cipher suites as well as certain TLS 1.2 suites. Suites utilizing SHA1 are considered insecure.

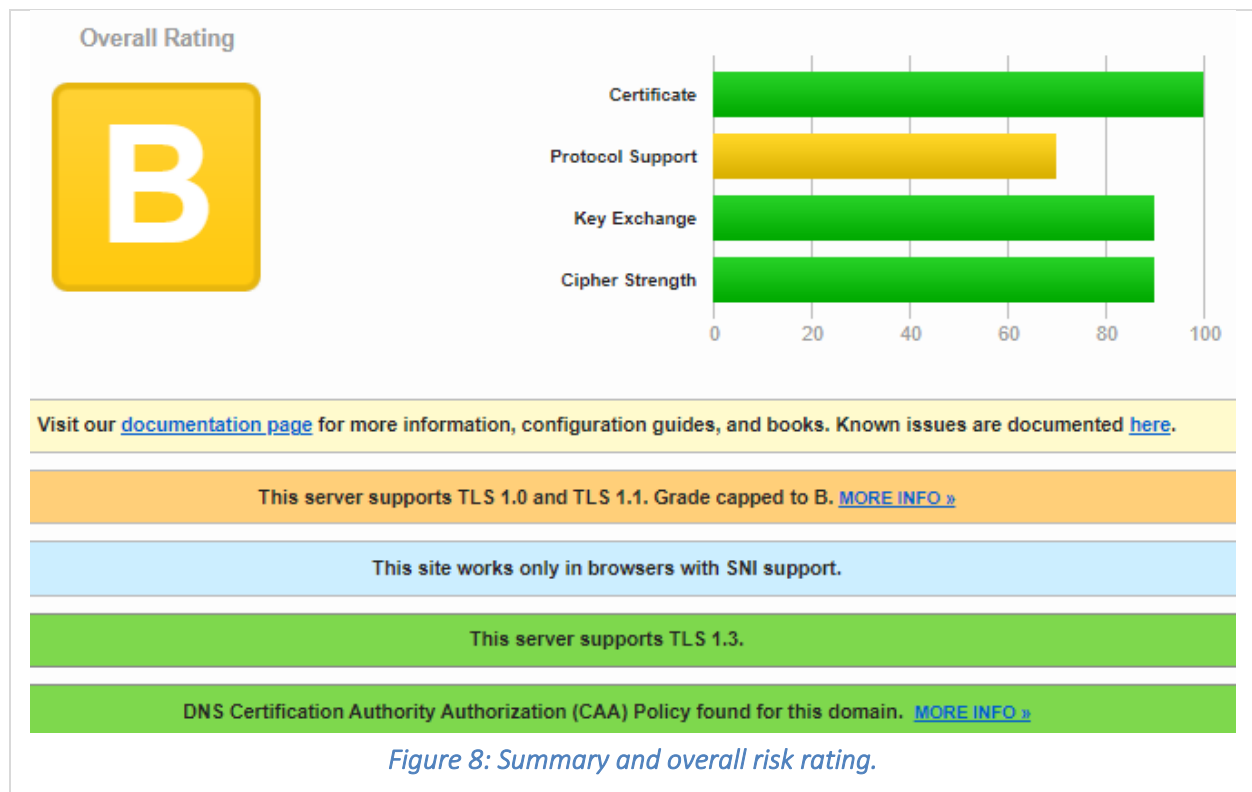
For additional information, please reference:



- [https://cheatsheetseries.owasp.org/cheatsheets/Transport Layer Protection Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport%20Layer%20Protection%20Cheat%20Sheet.html)

Test your website's TLS/SSL and cipher suites using the following tool to analyze your current configuration:

- <https://www.ssllabs.com/ssltest/index.html>

Evidence

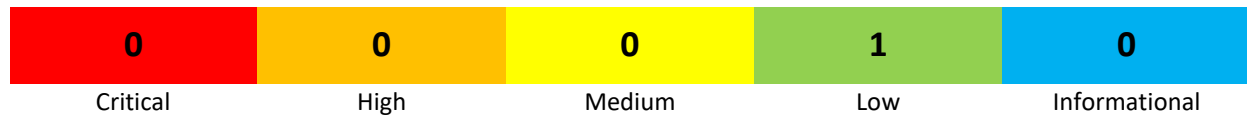


# TLS 1.2 (suites in server-preferred order)			
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)	ECDH x25519 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0a9)	ECDH x25519 (eq. 3072 bits RSA) FS		256 ^P
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)	ECDH x25519 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH x25519 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0a8)	ECDH x25519 (eq. 3072 bits RSA) FS		256 ^P
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	ECDH x25519 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)		WEAK	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)		WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		WEAK	112
# TLS 1.1 (suites in server-preferred order)			
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		WEAK	112
# TLS 1.0 (suites in server-preferred order)			
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		WEAK	112

(P) This server prefers ChaCha20 suites with clients that don't have AES-NI (e.g., Android devices)

Figure 9: The application uses TLS 1.0 and TLS 1.1, which have been deprecated, as well as multiple weak TLS 1.2 cipher suites.

4.2. CARE360.CAREVALIDATE.COM (18.205.36.100)

**care360.carevalidate.com (18.205.36.100)**

Platform	Ionic, Heroku
Server	Nginx
Port(s)	80, 443
Path	/

Severity Score	Description
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Usage of Weak Cipher Suites The server accepts weak cipher suites with known vulnerabilities that affect integrity and confidentiality.

4.2.1. USAGE OF WEAK CIPHER SUITES (CVSS: 3.7)

CVSS Score	Category	Type	Required Access
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Configuration	Use of a Broken or Risky Cryptographic Algorithm	Network
Risk Evaluation	CWE ID	CAPEC ID	OWASP Top 10
Medium to Low	<u>CWE-327</u>	<u>CAPEC-97</u>	<u>A02:2021-Cryptographic Failures</u>
Location			
Within the following TLS 1.2 cipher suites: <ul style="list-style-type: none">• TLS_RSA_WITH_AES_128_GCM_SHA256• TLS_RSA_WITH_AES_256_GCM_SHA384			
Description			
<p>Cryptographic protocols such as SSL and TLS allow for the safe transmission of information between two parties, ensuring that all sensitive information and communication remain unmodified and private. However, the usage of weak ciphers may allow an adversary to read or maliciously alter the data between the two points of communication through a Machine-in-the-Middle (MitM) attack.</p> <p>Cipher suites with any of the following should be avoided:</p> <ul style="list-style-type: none">• CBC• DES/3DES• RC4• Key lengths under 112 bits			
Impact			
An adversary may be able to conduct a MitM attack and sniff or modify sensitive information such as login credentials and user emails.			

Likelihood

Fully exploiting this vulnerability requires that an adversary be placed between the communication channel between the victim and the server. The victim is actively communicating with the vulnerable server, which significantly increases the difficulty of exploit.

Remediation

- Remove all weak cryptographic ciphers from the list of accepted cipher suites.
- Disable SSL usage, and only utilize cipher suites currently considered secure. These would include all TLS 1.3 cipher suites as well as certain TLS 1.2 suites. Suites utilizing SHA1 are considered insecure.

For additional information, please reference:

- [https://cheatsheetseries.owasp.org/cheatsheets/Transport Layer Protection Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport%20Layer%20Protection%20Cheat%20Sheet.html)

Test your website's TLS/SSL and cipher suites using the following tool to analyze your current configuration:

- <https://www.ssllabs.com/ssltest/index.html>

Evidence

Summary

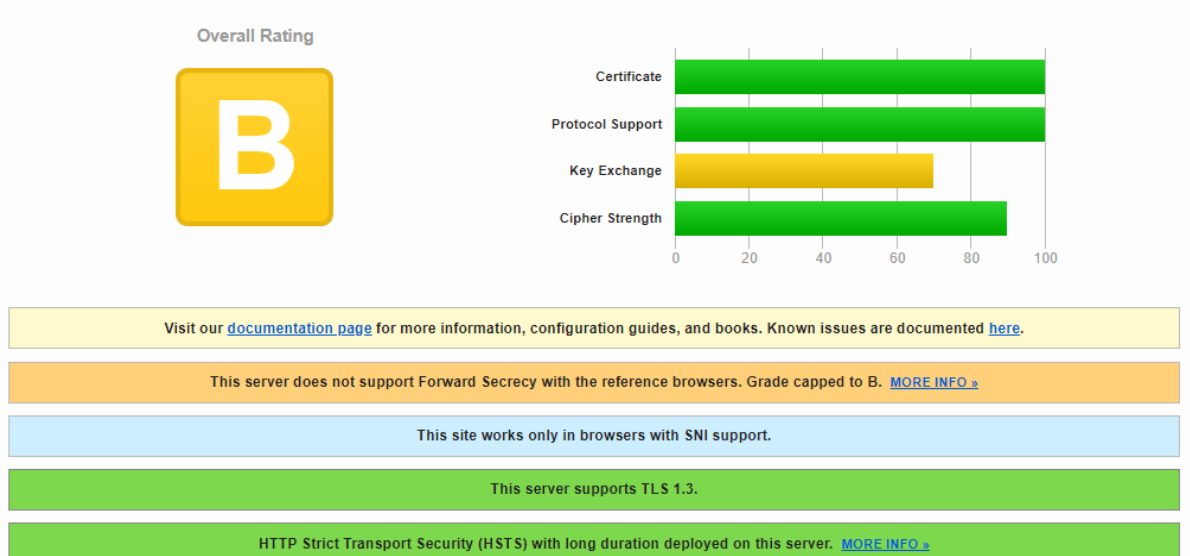


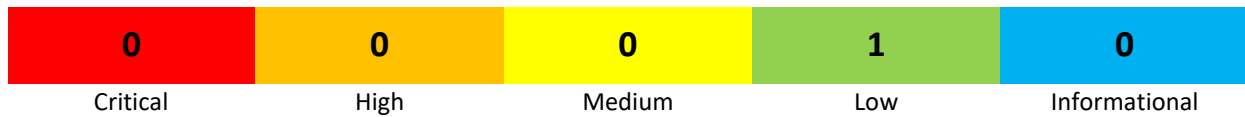
Figure 10: Summary and overall risk rating.

TLS 1.2 (suites in server-preferred order)

TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc030)	ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH secp256r1 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK		128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) WEAK		256

Figure 11: The application accepts weak TLS 1.2 communications; Additionally, these ciphers do not utilize Forward Secrecy.

4.3. CONTACT-TRACING-PROD.APPSPOT.COM (142.251.45.52)

**contact-tracing-prod.appspot.com (142.251.45.52)**

Platform	React, Node.js
Server	Express
Port(s)	80, 443
Path	/

Severity Score	Description
3.7 <small>(AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)</small>	Usage of Weak Cipher Suites The server accepts weak cipher suites with known vulnerabilities that affect integrity and confidentiality.

4.3.1. USAGE OF WEAK CIPHER SUITES (CVSS: 3.7)

CVSS Score	Category	Type	Required Access
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Configuration	Use of a Broken or Risky Cryptographic Algorithm	Network
Risk Evaluation	CWE ID	CAPEC ID	OWASP Top 10
Medium to Low	<u>CWE-327</u>	<u>CAPEC-97</u>	<u>A02:2021-Cryptographic Failures</u>
Location			
Within the following TLS 1.0, TLS 1.1, and TLS 1.2 cipher suites:			
<ul style="list-style-type: none">• TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA• TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA• TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA• TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA• TLS_RSA_WITH_AES_128_GCM_SHA256• TLS_RSA_WITH_AES_256_GCM_SHA384• TLS_RSA_WITH_AES_128_CBC_SHA• TLS_RSA_WITH_AES_256_CBC_SHA• TLS_RSA_WITH_3DES_EDE_CBC_SHA			
Description			
<p>Cryptographic protocols such as SSL and TLS allow for the safe transmission of information between two parties, ensuring that all sensitive information and communication remain unmodified and private. However, the usage of weak ciphers may allow an adversary to read or maliciously alter the data between the two points of communication through a Machine-in-the-Middle (MitM) attack.</p> <p>Cipher suites with any of the following should be avoided:</p>			

- CBC
- DES/3DES
- RC4
- Key lengths under 112 bits

Impact

An adversary may be able to conduct a MitM attack and sniff or modify sensitive information such as login credentials and user emails.

Likelihood

Fully exploiting this vulnerability requires that an adversary be placed between the communication channel between the victim and the server. The victim is actively communicating with the vulnerable server, which significantly increases the difficulty of exploit.

Remediation

- Remove all weak cryptographic ciphers from the list of accepted cipher suites.
- Disable SSL usage, and only utilize cipher suites currently considered secure. These would include all TLS 1.3 cipher suites as well as certain TLS 1.2 suites. Suites utilizing SHA1 are considered insecure.

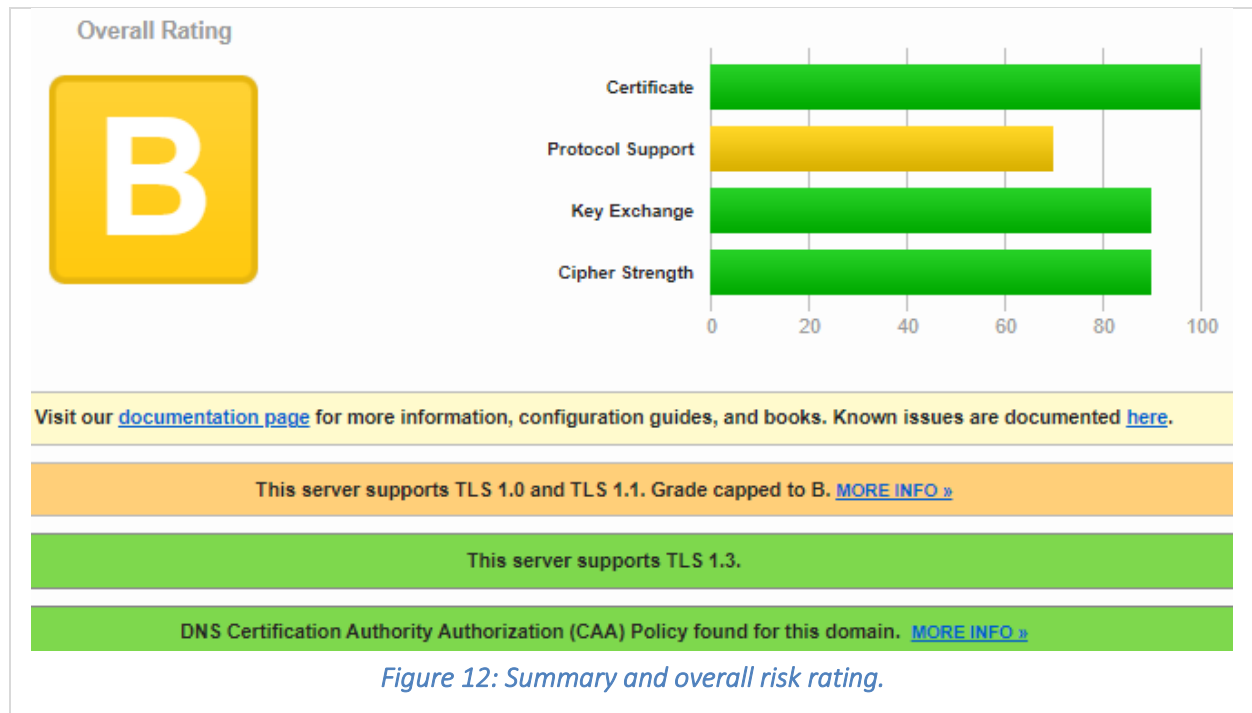
For additional information, please reference:

- [https://cheatsheetseries.owasp.org/cheatsheets/Transport Layer Protection Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport%20Layer%20Protection%20Cheat%20Sheet.html)

Test your website's TLS/SSL and cipher suites using the following tool to analyze your current configuration:

- <https://www.ssllabs.com/ssltest/index.html>

Evidence



# TLS 1.2 (suites in server-preferred order)			[-]
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)	ECDH x25519 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)	ECDH x25519 (eq. 3072 bits RSA) FS		256 ^P
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)	ECDH x25519 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH x25519 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)	ECDH x25519 (eq. 3072 bits RSA) FS		256 ^P
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	ECDH x25519 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)		WEAK	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)		WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		WEAK	112
# TLS 1.1 (suites in server-preferred order)			[-]
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		WEAK	112
# TLS 1.0 (suites in server-preferred order)			[-]
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	ECDH x25519 (eq. 3072 bits RSA) FS	WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		WEAK	112
(P) This server prefers ChaCha20 suites with clients that don't have AES-NI (e.g., Android devices)			

Figure 13: The application uses TLS 1.0 and TLS 1.1, which have been deprecated, as well as multiple weak TLS 1.2 cipher suites.

4.4. API.CARE360.CAREVALIDATE.COM (18.205.222.128)

0	0	0	1	0
Critical	High	Medium	Low	Informational

api.care360.carevalidate.com (18.205.222.128)

Platform	Heroku
Server	Cowboy
Port(s)	80, 443
Path	/

Severity Score	Description
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Usage of Weak Cipher Suites The server accepts weak cipher suites with known vulnerabilities that affect integrity and confidentiality.

4.4.1. USAGE OF WEAK CIPHER SUITES (CVSS: 3.7)

CVSS Score	Category	Type	Required Access
3.7 (AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N)	Configuration	Use of a Broken or Risky Cryptographic Algorithm	Network
Risk Evaluation	CWE ID	CAPEC ID	OWASP Top 10
Medium to Low	<u>CWE-327</u>	<u>CAPEC-97</u>	<u>A02:2021-Cryptographic Failures</u>
Location			
Within the following TLS 1.2 cipher suites: <ul style="list-style-type: none">TLS_RSA_WITH_AES_128_GCM_SHA256			

- TLS_RSA_WITH_AES_256_GCM_SHA384

Description

Cryptographic protocols such as SSL and TLS allow for the safe transmission of information between two parties, ensuring that all sensitive information and communication remain unmodified and private. However, the usage of weak ciphers may allow an adversary to read or maliciously alter the data between the two points of communication through a Machine-in-the-Middle (MitM) attack.

Cipher suites with any of the following should be avoided:

- CBC
- DES/3DES
- RC4
- Key lengths under 112 bits

Impact

An adversary may be able to conduct a MitM attack and sniff or modify sensitive information such as login credentials and user emails.

Likelihood

Fully exploiting this vulnerability requires that an adversary be placed between the communication channel between the victim and the server. The victim is actively communicating with the vulnerable server, which significantly increases the difficulty of exploit.

Remediation

- Remove all weak cryptographic ciphers from the list of accepted cipher suites.
- Disable SSL usage, and only utilize cipher suites currently considered secure. These would include all TLS 1.3 cipher suites as well as certain TLS 1.2 suites. Suites utilizing SHA1 are considered insecure.

For additional information, please reference:

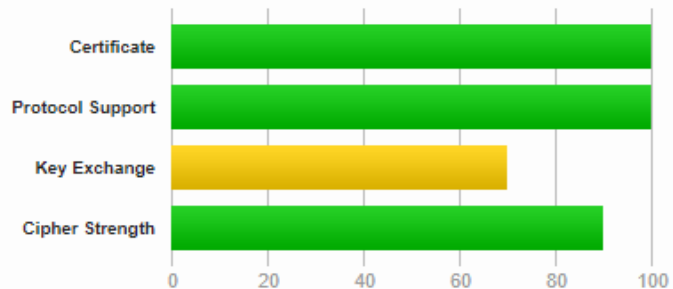
- https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

Test your website's TLS/SSL and cipher suites using the following tool to analyze your current configuration:

- <https://www.ssllabs.com/ssltest/index.html>

Evidence

Overall Rating



Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

This server does not support Forward Secrecy with the reference browsers. Grade capped to B. [MORE INFO »](#)

This site works only in browsers with SNI support.

This server supports TLS 1.3.

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. [MORE INFO »](#)

Figure 14: Summary and overall risk rating.

Cipher Suites

# TLS 1.3 (suites in server-preferred order)				
TLS_AES_128_GCM_SHA256 (0x1301)	ECDH x25519 (eq. 3072 bits RSA)	FS		128
TLS_AES_256_GCM_SHA384 (0x1302)	ECDH x25519 (eq. 3072 bits RSA)	FS		256
TLS_CHACHA20_POLY1305_SHA256 (0x1303)	ECDH x25519 (eq. 3072 bits RSA)	FS		256 ^P
# TLS 1.2 (suites in server-preferred order)				
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)	ECDH secp256r1 (eq. 3072 bits RSA)	FS		256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	ECDH secp256r1 (eq. 3072 bits RSA)	FS		128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	ECDH secp256r1 (eq. 3072 bits RSA)	FS		256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)		WEAK		128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d)		WEAK		256

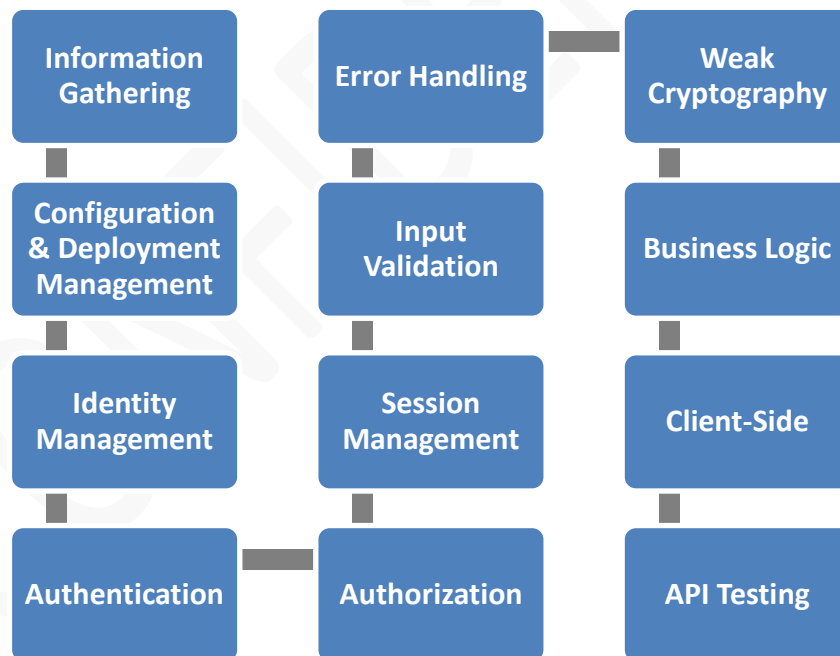
Figure 15: The application accepts weak TLS 1.2 communications; Additionally, these ciphers do not utilize Forward Secrecy.

5. METHODOLOGY AND STANDARDS

5.1. OWASP WEB SECURITY TESTING GUIDE (WSTG)



The OWASP Web Security Testing Guide (WSTG) is a comprehensive methodology for testing the security of web applications. It aims to provide full coverage of possible vulnerabilities and is used in a way that is consistent, reproducible, rigorous, and under quality control. The WSTG combines methods, techniques, tools, and resources for testing a wide array of web application security issues.



The WSTG splits the penetration testing into 12 sub-categories for a total of 97 controls:

- **Information Gathering:** Gathering information about the target web application, e.g., the technology used, version numbers, and any available public information. This information is used to identify potential vulnerabilities.

- **Configuration and Deployment Management:** Testing the security of the web server, server configurations, and the deployment environment. This includes checking for misconfigurations and default settings that could leave the application vulnerable.
- **Identity Management:** Testing the identity management mechanisms, including the management of user roles, account provisioning, and user registration weaknesses.
- **Authentication:** Testing the authentication mechanism to identify any weaknesses that could be exploited to gain unauthorized access.
- **Authorization:** Testing the authorization mechanism to identify any flaws that could be exploited to gain access to restricted resources.
- **Session Management:** Testing the session management system to identify any vulnerabilities that could be exploited to hijack user sessions.
- **Input Validation:** Testing the input validation mechanism to identify any weaknesses that could be exploited to inject malicious data into the application.
- **Error Handling:** Testing the error handling mechanism to identify any information leaks or other weaknesses that could be exploited.
- **Weak Cryptography:** Testing the encryption mechanisms used by the application to identify any weaknesses that could be exploited.
- **Business Logic:** Testing the business logic of the application to identify any vulnerabilities that could be exploited to gain unauthorized access or to manipulate data.
- **Client-Side:** Testing the client-side code, such as JavaScript, to identify any vulnerabilities that could be exploited to compromise the security of the application.
- **API Testing:** Testing the APIs used by the application to communicate with other systems or services.

Both manual and automated testing techniques were used in each of these categories to ensure a comprehensive and effective testing process. This approach helps to identify a broader range of vulnerabilities, providing a more accurate picture of the application's security posture.

For additional information, please refer to the following link:

- <https://owasp.org/www-project-web-security-testing-guide/>

5.2. RISK RATING METHODOLOGY

Using the OWASP Risk Rating Methodology as a reference, Websec has incorporated the following factors into calculating the risk of each vulnerability:

- Ease of Discoverability
- Loss of Confidentiality
- Loss of Availability
- Ease of Exploitation
- Loss of Integrity

Other factors such as Awareness, Intrusion Detection, and Loss of Accountability have been excluded from the Risk Rating due to a lack of information required to conduct a full risk analysis. For every encountered vulnerability, a score is assigned for each metric from which the average is taken and used to estimate the risk of the given vulnerability.

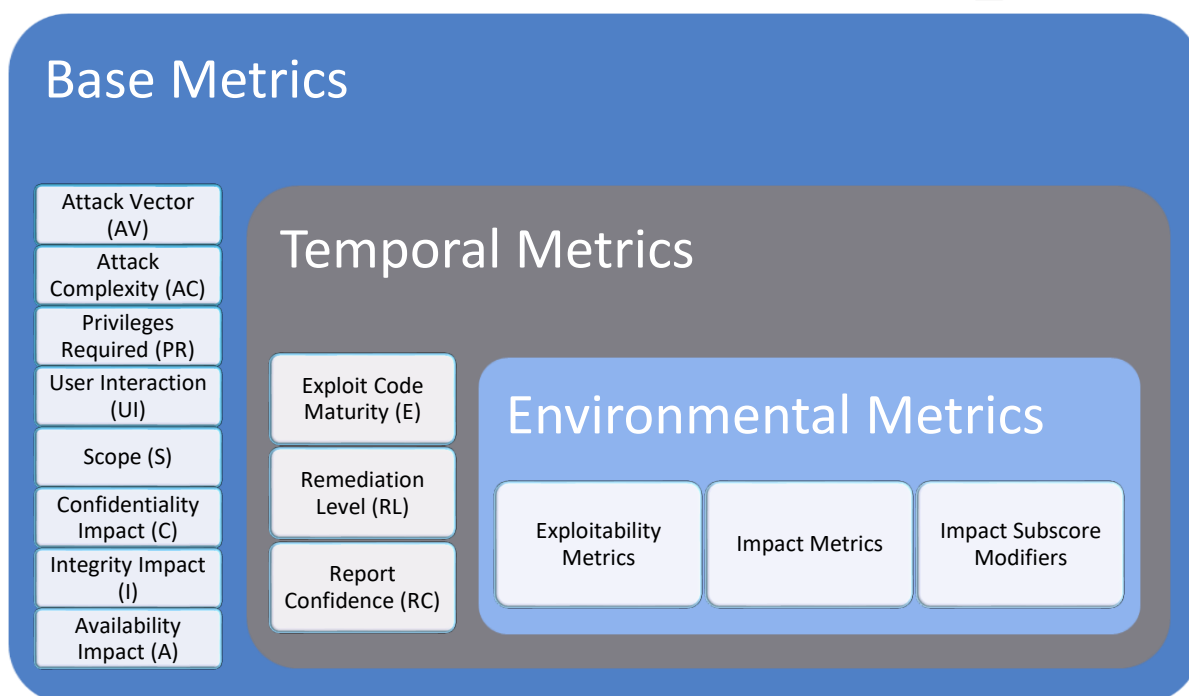
LIKELIHOOD AND IMPACT LEVELS	
SEVERITY	CVSS SCORE
Critical	9.0 to 10.0
High	7.0 to 8.9
Medium	4.0 to 6.9
Low	0.1 to 3.9
Informational	N/A

For additional information, refer to:

- https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

5.3. COMMON VULNERABILITY SCORING SYSTEM (CVSS)

The ability to assess the risk associated with each vulnerability is extremely important in the line of vulnerability analysis and risk management. The CVSS is an industry standard with the purpose of analyzing the fundamental characteristics of vulnerabilities to assign a value to each vulnerability and provide users with a clear and objective understanding of the risk involved. The system has three groups of metrics, each dependent on the previous one:



- **Base Metrics** represents the intrinsic and fundamental characteristics of a vulnerability that are constant with time and across user environments. The Access Vector, Access Complexity, and Authentication metrics capture how the vulnerability is accessed and whether extra conditions are required to exploit it. The three-impact metrics measure how a vulnerability, if exploited, will directly affect an IT asset, where the impacts are independently defined as the degree of loss of confidentiality, integrity, and availability.
- **Temporal Metrics** represents the impact and risk associated with a vulnerability over time as these may change over time. Three such factors that CVSS captures are: confirmation of the technical details of a vulnerability, the remediation status of the vulnerability, and the availability of exploit code or techniques. Each of these factors is important in the adjustment of urgency (i.e., the priority) of a vulnerability over time.

- **Environmental Metrics** represents the change in impact and risk due to the environment. Different environments can have an immense bearing on the risk that a vulnerability poses to an organization and its stakeholders. The CVSS environmental metric group captures the characteristics of a vulnerability that are associated with a user's IT environment.

5.3.1. VULNERABILITY SEVERITY RATINGS

The severity ratings are provided by the National Vulnerability Database (NVD). The severity rankings of “Low”, “Medium”, “High”, and “Critical” that are mapped to the numeric CVSS scores.

CVSS V3 Ratings:

- Vulnerabilities are labeled “**Low**” severity if they have a base score of 0.1 – 3.9.
- Vulnerabilities are labeled “**Medium**” severity if they have a base score of 4.0 – 6.9.
- Vulnerabilities are labeled “**High**” severity if they have a base score of 7.0 – 8.9.
- Vulnerabilities are labeled “**Critical**” severity if they have a base score of 9.0 – 10.0.

For a complete guide on the CVSS, please refer to:

- <https://www.first.org/cvss/v3.1/user-guide>

5.4. COMMON WEAKNESS ENUMERATION (CWE)



Common Weakness Enumeration (CWE) is a list of Weakness Types (“Vulnerabilities”) in software for developers and security professionals. The CWE’s purpose is to unify the description of software security weaknesses in architecture, design, and code. It can be seen as a catalogue of documented faults that often occur in programming and could lead to vulnerabilities. The CWE is widely used by different security tools responsible for identifying weaknesses and promoting vulnerability identification, mitigation, and prevention.

The CWE addresses the needs of large companies and organizations to become aware of and catalogue various weaknesses. The CWE provides a common language for issues, a metric for security testing, and a baseline for weakness identification. Consumers expect their products and solutions, whether purchased or contracted, to be protected appropriately against known weaknesses. If the software is developed with CWE in mind, the chances of suffering from vulnerabilities are highly reduced.

The Common Vulnerabilities and Exposures (CVE) is a reference system for publicly disclosed security concerns. Much like the CWE, the CVE is used to catalogue vulnerabilities; however, the CVE focuses on vulnerabilities in specific software, while the CWE addresses vulnerabilities in general by type.

The CWE and CVE are initiatives of the MITRE Corporation, sponsored by the US National Cybersecurity FFRDC, US-CERT, and Homeland Security. These systems draw from diverse viewpoints throughout the industry with the broad goal of helping secure academia, businesses, and government.



WEBSEC

info@websec.ca

—

Websec Information Security Services, Inc.

Victoria, British Columbia, Canada

websec.ca

