Sri Lanka Institute of Information Technology

IT21299902 (ZAKEY M.S.M.A)

Bug bounty Report 02.

Domain: MalwareBytes.com

Web security – IE2062

B.Sc. (Hons) in Information Technology Specialization in cyber security.

Declaration:

- I hereby certify that no part of this assignment has been copied from any other work or any other source. No part of this assignment has been written/produced for me by another person.

- I hold a copy of this assignment that I can produce if the original is lost or damaged.

1. Sensitive Data exposure.

- **Weak ciphers enabled.**

I hope this message finds you well. I am writing to report a vulnerability I discovered under the domain of https://www.malwarebytes.com during my participation in the bug bounty program. Please find below the details of the vulnerability for your review and further action.

Vulnerability title: Weak ciphers enabled.

Severity: Medium

OWASP classification 2013: A6

CVSS 3.0 score: 6.8

CVSS 3.1 score: 6.8

CVSS string: CVSS:3.1/AV: A/AC:H/PR: N/UI: N/S: U/C:H/I:H/A: N

Effected components: SSL/TLS communication.

Authentication.

PCI/DSS security frameworks.

Date of Discovery: 2023/05/10

Date of Report: 2023/05/12

2. Vulnerability Description.

I have identified a weak cipher enabled vulnerability within the affected system. This vulnerability allows an attacker to do Cipher suite downgrade, Eavesdropping, man-in-the-middle, or data tempering attacks. By exploiting this vulnerability, an attacker could potentially steal sensitive user information, manipulate website content, or perform other malicious activities.

3. Impact assessment.

Under this OWASP category I have identified many vulnerabilities that can occur due to bad implementation of security implementations under domain of **https://www.malwarebytes.com/**

1. So, the effect of allowing weak ciphers during SSL communication is attackers might decrypt the SSL traffic between client and server if they used good mechanism.

2. The online application is more vulnerable to numerous cryptographic attacks with weak ciphers, such as brute force, cipher-text-only, and chosen-plaintext. Attackers can use these flaws to decrypt private information sent between the client and the server.

3. Security requirements and laws may not be followed if weak ciphers are enabled. Strong cryptographic protocols and ciphers are necessary to secure sensitive data according to many security standards, including the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS). If these criteria are not met, there may be legal and regulatory repercussions.

4. In comparison to stronger ciphers, some weaker ciphers could have a higher computational cost. The server's processing time and resource use may rise if such ciphers are enabled, which can affect the web application's performance and response times.

Because of this vulnerability marked as MEDIUM. It's a must to address it with a good solution.

This domain supports following weak ciphers:

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027

4. Steps to Reproduce.

1. Must first find the list of ciphers that your web application supports. The SSL/TLS settings on the server are normally setup with this data.

2. There are several internet resources that can scan the SSL/TLS settings of your web application and reveal the ciphers that are enabled. The SSL Server Test from SSL Labs is one such tool **(https://www.ssllabs.com/ssltest/)**. The tool will conduct a thorough examination of your SSL/TLS configuration, including the supported ciphers, after you enter the URL of your web application**. (http://www.malwarebytes.com)**

3. Review the scan results when the program has finished its study to find any ciphers that are enabled in your web application that are weak or out-of-date. The program will often give the SSL/TLS configuration a grade or score and point out any flaws or vulnerabilities it discovers.

4. Test cipher negotiation: Run a cipher negotiation test against your web application using a network testing tool like OpenSSL or Nmap. In this test, you'll establish a connection to your web application and watch as the negotiated cipher suite is established during the handshake.

5. Examine the cipher suite that was agreed upon: See if any weak ciphers were chosen for the connection by looking at the results of the encryption negotiation test. Search for any ciphers that are regarded as being brittle, exposed, or dated.

6. Test on various clients: To watch the SSL/TLS handshake, access your web application using various web browsers and client devices. Verify whether weak ciphers are consistently chosen and whether the negotiated cipher suite varies between clients.

7. Validate with vulnerability scanning tools: Conduct a thorough security analysis of your web application using vulnerability scanning tools like Nessus or OpenVAS. These programs may check for SSL/TLS vulnerabilities and offer thorough findings on ciphers with weak security or possible attacks.

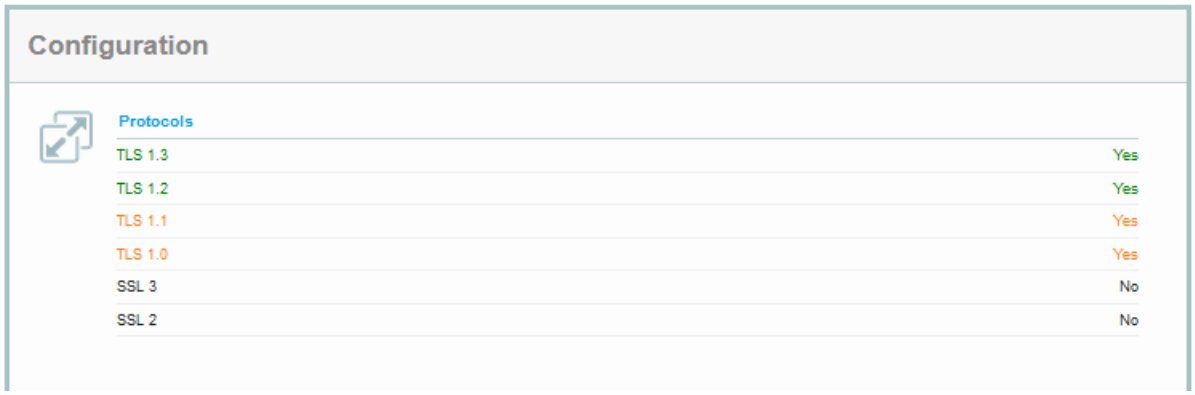
5. Proof of concept.

1. SSL/TLS scan report by SSLLabs

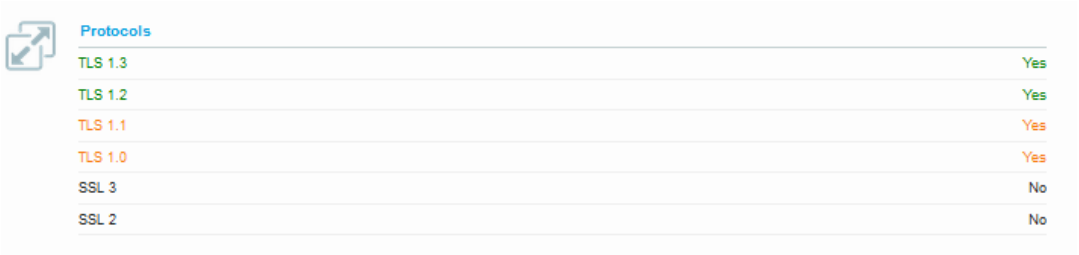
Certificate #1: RSA 2048 bits (SHA256withRSA)

Server Key and Certificate #1

Subject	malwarebytes.com Fingerprint SHA256: 75cdc223af69e2c49a9a8fd11e6040960513c71e77883413f21285d7a43a8b1b Pin SHA256: FYc3OTCEQQcxLPEyAcMNES7dNwMHZLrD/ICXsfkoWbQ=
Common names	malwarebytes.com
Alternative names	malwarebytes.com *.beta.malwarebytes.com *.cloud.malwarebytes.com *.mbamupdates.com *.api.cloud.malwarebytes.com *.malwarebytes.com *.api-stage.cloud.malwarebytes.com *.sre.malwarebytes.com *.malwarebytes.org *.data.service.malwarebytes.org *.mwbsys.com *.eng-prod.mb-internal.com *.mb-cosmos.com
Serial Number	0a018051b91c334887cb4179d526bef1
Valid from	Thu, 23 Feb 2023 00:00:00 UTC
Valid until	Fri, 03 Nov 2023 23:59:59 UTC (expires in 5 months and 9 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	Amazon RSA 2048 M01 AIA: http://crt.r2m01.amazontrust.com/r2m01.cer
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl.r2m01.amazontrust.com/r2m01.crl OCSP: http://ocsp.r2m01.amazontrust.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

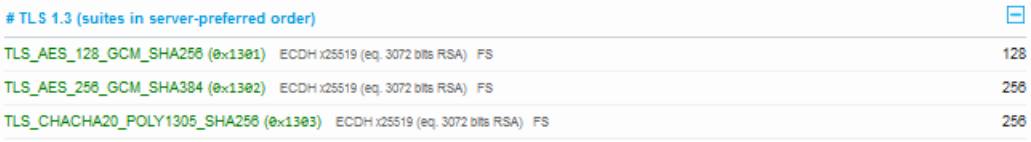


Used protocols.

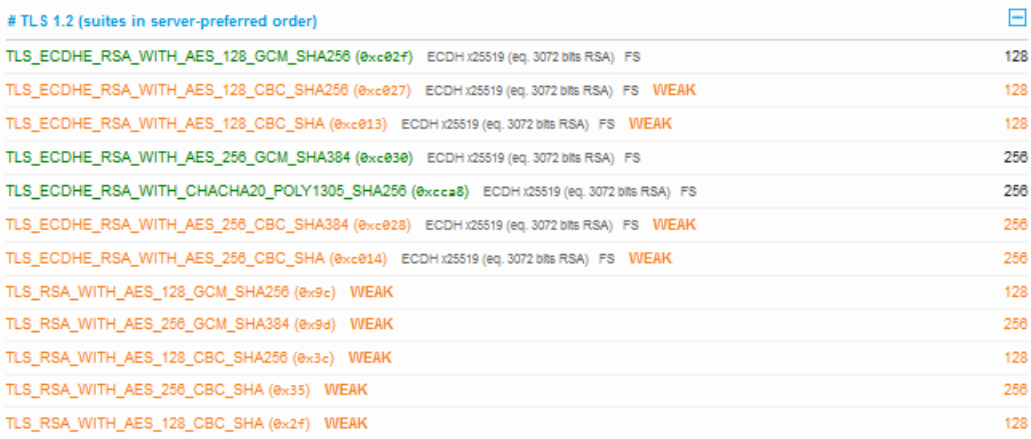


Supported cipher suites in each protocol.

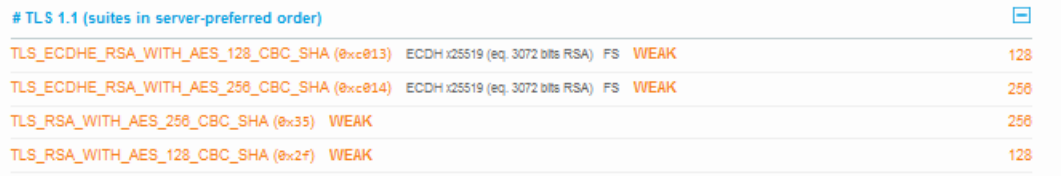
1. TLS 1.3



2. TLS 1.2

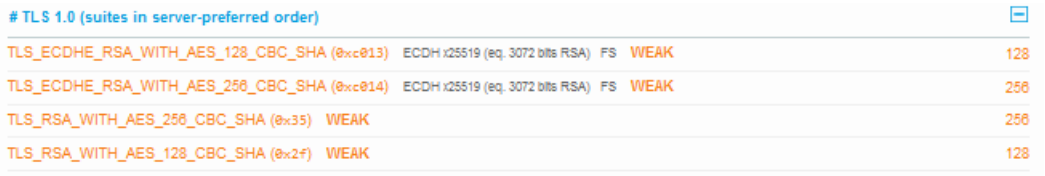


3. TLS 1.1



# TLS 1.1 (suites in server-preferred order)			
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_256_CBC_SHA (0x35)	WEAK		256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	WEAK		128

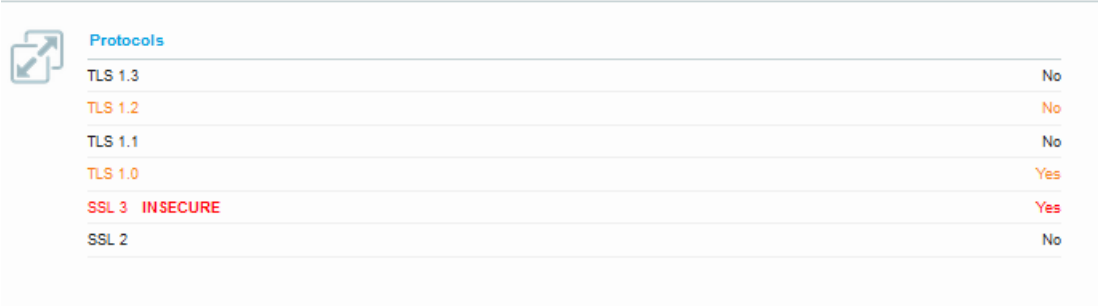
4. TLS 1.0



# TLS 1.0 (suites in server-preferred order)			
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_256_CBC_SHA (0x35)	WEAK		256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)	WEAK		128

SSL/TLS handshake simulation in different clients.

1. Android 4.0.4



Protocols	
TLS 1.3	No
TLS 1.2	No
TLS 1.1	No
TLS 1.0	Yes
SSL 3 INSECURE	Yes
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) WEAK	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) WEAK	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) WEAK	256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x38) WEAK	256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f) WEAK	256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005) WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) WEAK	112
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc008) WEAK	112
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x16) WEAK	112
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) WEAK	112
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA (0xc00d) WEAK	112
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc003) WEAK	112
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) WEAK	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) WEAK	128
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x32) WEAK	128
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xc00e) WEAK	128
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004) WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011) INSECURE	128
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007) INSECURE	128
TLS_ECDH_RSA_WITH_RC4_128_SHA (0xc00c) INSECURE	128
TLS_ECDH_ECDSA_WITH_RC4_128_SHA (0xc002) INSECURE	128
TLS_RSA_WITH_RC4_128_SHA (0x5) INSECURE	128
TLS_RSA_WITH_RC4_128_MD5 (0x4) INSECURE	128
TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0xff)	-

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	Yes INSECURE
Session tickets	Yes
OCSP stapling	No
Signature algorithms	-
Named Groups	sect163k1, sect163r1, sect163r2, sect193r1, sect193r2, sect233k1, sect233r1, sect239k1, sect283k1, sect283r1, sect409k1, sect409r1, sect571k1, sect571r1, secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, secp224k1, secp224r1, secp256k1, secp256r1, secp384r1, secp521r1
Next Protocol Negotiation	Yes
Application Layer Protocol Negotiation	No
SSL 2 handshake compatibility	No

2. Android 4.3

Protocols

TLS 1.3	No
TLS 1.2	No
TLS 1.1	No
TLS 1.0	Yes
SSL 3 INSECURE	Yes
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) WEAK	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) WEAK	256
TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA (0xc022) WEAK	256
TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA (0xc021) WEAK	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) WEAK	256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x38) WEAK	256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f) WEAK	256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005) WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) WEAK	112
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc008) WEAK	112
TLS_SRP_SHA_DSS_WITH_3DES_EDE_CBC_SHA (0xc01c) WEAK	112
TLS_SRP_SHA_RSA_WITH_3DES_EDE_CBC_SHA (0xc01b) WEAK	112
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x16) WEAK	112
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) WEAK	112
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA (0xc00d) WEAK	112
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc003) WEAK	112
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) WEAK	128
TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA (0xc01f) WEAK	128
TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA (0xc01e) WEAK	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) WEAK	128
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x32) WEAK	128
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xc00e) WEAK	128
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004) WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011) INSECURE	128
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007) INSECURE	128
TLS_ECDH_RSA_WITH_RC4_128_SHA (0xc00c) INSECURE	128
TLS_ECDH_ECDSA_WITH_RC4_128_SHA (0xc002) INSECURE	128
TLS_RSA_WITH_RC4_128_SHA (0x5) INSECURE	128
TLS_RSA_WITH_RC4_128_MD5 (0x4) INSECURE	128
TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0xff)	-

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	No
Signature algorithms	-
Named Groups	sect571r1, sect571k1, secp521r1, sect409k1, sect409r1, secp384r1, sect283k1, sect283r1, secp256k1, secp256r1, sect239k1, sect233k1, sect233r1, secp224k1, secp224r1, sect193r1, sect193r2, secp192k1, secp192r1, sect163k1, sect163r1, sect163r2, secp160k1, secp160r1, secp160r2
Next Protocol Negotiation	Yes
Application Layer Protocol Negotiation	No
SSL 2 handshake compatibility	No

3. Chorme 80 /windows 10

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_GREASE_4A (0x4a4a)	-
TLS_AES_128_GCM_SHA256 (0x1301) Forward Secrecy	128
TLS_AES_256_GCM_SHA384 (0x1302) Forward Secrecy	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) 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

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/ECDSA, RSA_PSS_SHA256, SHA256/RSA, SHA384/ECDSA, RSA_PSS_SHA384, SHA384/RSA, RSA_PSS_SHA512, SHA512/RSA, SHA1/RSA
Named Groups	tls_grease_0a0a, x25519, secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

4. IE 8 / XP

Protocols

TLS 1.3	No
TLS 1.2	No
TLS 1.1	No
TLS 1.0	Yes
SSL 3 INSECURE	Yes
SSL 2	No

Cipher Suites (in order of preference)

TLS_RSA_WITH_RC4_128_MD5 (0x4) INSECURE	128
TLS_RSA_WITH_RC4_128_SHA (0x5) INSECURE	128
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112
TLS_RSA_WITH_DES_CBC_SHA (0x9) INSECURE	56
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x64) INSECURE	56
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x62) INSECURE	56
TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x3) INSECURE	40
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x6) INSECURE	40
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) WEAK	112
TLS_DHE_DSS_WITH_DES_CBC_SHA (0x12) INSECURE	56
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x63) INSECURE	56

Protocol Details	
Server Name Indication (SNI)	No
Secure Renegotiation	Yes
TLS compression	No
Session tickets	No
OCSP stapling	No
Signature algorithms	-
Named Groups	-
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	No
SSL 2 handshake compatibility	No

Testing vulnerabilities related to weak cipher suits.

Protocol Details	
DROWN	Unable to perform this test due to an internal error. (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete INTERNAL ERROR: connect timed out INTERNAL ERROR: connect timed out
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Not mitigated server-side (more info) TLS 1.0: 0xc013
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info) TLS 1.2 : 0xc027
GOLDENDOODLE	No (more info) TLS 1.2 : 0xc027
OpenSSL 0-Length	No (more info) TLS 1.2 : 0xc027
Sleeping POODLE	No (more info) TLS 1.2 : 0xc027

Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	With modern browsers (more info)
ALPN	Yes h2 http/1.1
NPN	No
Session resumption (caching)	No (IDs assigned but not accepted)
Session resumption (tickets)	Yes
OCSP stapling	Yes
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	Unknown
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No, DHE suites not supported
DH public server param (Ys) reuse	No, DHE suites not supported
ECDH public server param reuse	No
Supported Named Groups	x25519, secp256r1, secp384r1 (server preferred order)
SSL 2 handshake compatibility	No
0-RTT enabled	No

6. Proposed mitigation/fix.

Solution may differ mechanism used to implement the cipher suite. Can suggest some solutions for Apache HTTP server, Nginx, Microsoft IIS, and node js.

1. Apache

Change the SSLCipherSuite directive in the Apache configuration file (such as httpd.conf or ssl.conf) to only include strong ciphers. Take out of the list any ciphers that are old or weak.

```
SSLCipherSuite HIGH:!aNULL:!MD5:!3DES
```

2. Nginx

Update the ssl_ciphers directive in the Nginx configuration file (such as nginx.conf or ssl.conf) to only contain secure ciphers. Eliminate any ciphers that are weak or exposed**.**

```
ssl_ciphers 'HIGH:!aNULL:!MD5:!3DES';
```

3. Node js

Your Node.js application's SSL/TLS configuration must be updated. Only safe ciphers should be included in the cipher's parameter of the https. create Server () function.

```
const https = require('https');
const fs = require('fs');

const options = {
  key: fs.readFileSync('path/to/private-key.pem'),
  cert: fs.readFileSync('path/to/certificate.pem'),
  ciphers: 'HIGH:!aNULL:!MD5:!3DES',
};

const server = https.createServer(options, (req, res) => {
  // Server logic
});

server.listen(443);
```

4. Microsoft IIS

For SSL/TLS configuration, use the IIS Manager. Go to your website's "SSL Settings" and, under "Ciphers," only choose strong ciphers. Cross out any weak ciphers from the list.