IMMUNIWEB SSL SECURITY TEST

API Documentation v 2.2.8

APRIL 1, 2024

Table of Contents

1. General Overview

API Documentation and How-To

API Specifications

Field Name	Value
Protocol	HTTPS
Request Type	POST
URLs	To initiate the test: https://www.immuniweb.com/ssl/api/v1/check/[ustamp].html To fetch the results: https://www.immuniweb.com/ssl/api/v1/get_result/[ustamp].html Where [ustamp] is an arbitrary UNIX time-stamp (must be an integer). Such construction is done to prevent caching on client side.

POST Specifications

Field Name	Value
domain	Must be a valid domain name, or IP address, followed by a port number. The port number is optional. If it is not supplied, 443 is used by default. Example: “domain=twitter.com:443”
show_test_results	"false" means that test results will be hidden, "true" means that test results will be displayed in Latest Tests section on ImmuniWeb’s website.
choosen_ip	IP address of tested server (if tested domain resolves to multiple addresses).
recheck	"false" will either use the results from the cache, or start a new test if the target has never been scanned. "true" will start a new test without checking the cache.
verbosity	“1” means that the output will be detailed and human-readable, “0” means output will be short.
token	The value of the token sent by the server if the tested domain is resolved into several IP addresses.
api_key	The optional secret token which you submit alongside with the request (used only for commercial access)

Example of a Transaction using cURL

Step 1: Starting the test

curl -d "domain=twitter.com:443&choosen_ip=any&show_test_results=true&recheck=false&verbosity=1" https://www.immuniweb.com/ssl/api/v1/check/1451425590.html

If you receive the “test_cached” status in the response, please proceed to Step 2.b

If you receive the “test_started” status in the response, please proceed to Step 2.a

Response Example (if the test has been found in the cache)


```
{
  "debug": {
    "recheck": "false",
    "hostname": "twitter.com",
    "choosen_ip": "104.244.42.65",
    "port": 443
  },
  "test_id": "cd9ee9ba7d1d6ba265c1f97bcabf954fa0c1eb84cf6727394d0ef6db536d0c58",
  "status": "test_cached",
  "status_id": 3,
  "message": "Test is cached"
}
```


Response Example (if the test has not been found in the cache):


```
{
  "debug": true,
  "job_id": "2a9e1f1bc92dc0c7a4bde930dff488771eea6d36988208d34163c5496227b8dc",
  "status": "test_started",
  "status_id": 1,
  "message": "Test has started"
}
```

Step 2.a: Fetching the results if the test was not found in the cache (call this until the test is finished)

curl -d "job_id=[job_id FROM STEP 1 RESPONSE]"
https://www.immuniweb.com/ssl/api/v1/get_result/1451425590.html

Response Example (if the test is not finished yet):

```
{
  "job_id": "0b8c7fdc3fbc97a0ffd309fea77651cc603e64bf710a83642bd442829492e8a9",
  "status": "in_progress",
  "status_id": 2,
  "eta": 2,
  "message": "Your test is in progress"
}
```

Step 2.b: Fetching the results if the test was found in the cache (“test_cached” status)

curl -d "id=[test_id FROM THE STEP 1 RESPONSE]"
https://www.immuniweb.com/ssl/api/v1/get_result/1451425590.html

Example with token

Step 1: Starting the test

curl -d "domain=twitter.com:443&show_test_results=true&recheck=false&verbosity=1"
https://www.immuniweb.com/ssl/api/v1/check/1451425590.html

Response Example:

```
{
  "multiple_ips": [
    "199.16.156.6",
    "199.16.156.102",
    "199.16.156.70",
    "199.16.156.230"
  ],
  "token": "68j3OCZLEomtjASxKoObjZXzX7p2M7L0"
}
```

Step 2: Starting the test with token

curl -d "domain=twitter.com:443&show_test_results=true&recheck=false&choosen_ip=199.16.156.230&verbosity=1&token=ABCD1234" https://www.immuniweb.com/ssl/api/v1/check/1451425590.html

Example with an error

curl -d "domain=0.0.0.0&show_test_results=true&recheck=false&verbosity=1" https://www.immuniweb.com/ssl/api/v1/check/1451425590.html

```
{
  "error": "The domain name cannot be resolved",
  "error_id": 7
}
```

How to download the PDF

curl -d "api_key=YOUR-API-KEY"
https://www.immuniweb.com/ssl/gen_pdf/[test_id]/ > report.pdf

The output will be composed of the following main elements that will be detailed later in this document:

Field Name	Type	Always present	Description
server_info	object	Yes	Contains basic server info, like IP, port, reverse DNS, etc.
certificates	array/ object	Yes	Contains information about certificates, graphs, etc.
nist	array/ object	Yes	Contains all information about NIST compliance
hipaa	array/ object	Yes	Contains all information about HIPAA compliance
pci_dss	array/ object	Yes	Contains all information about PCI DSS compliance
industry_best_ practices	array/ object	Yes	Containing information about industry best practices
email	array/ object	Yes	Containing all information about the mail server
third_party_ content	array/ object	Yes	Contains information about the third-party content
results	object	Yes	Contains the score and grade of the test
highlights	array/ object	Yes	Contains the highlights of the test
internals	object	Yes	Contains internal information such test IDs, server location, country, city and more.
page_title	string	Yes	Contains the page's title
company_details	object	No	Contains information about the company's name, country, state, city/locality. Syntax: `{ "company_details": { "country": "string", "state": "string", "name": "string", "locality": "string" }}`
third_party_ content_stats	object	No	Contains information about found external content – images, JS/CSS files, etc., and number of requests that failed. Syntax `{ "found": "int", "failed": "int" }`

2. Server Information

Server information part contains different elements about the server itself.

Please note: the syntax is as follows, unless specified otherwise:

```
{ "value": "string", "tag": "integer" }
```

Field Name	Type	Always present	Description
ip	object	Yes	The IP address tested.
port	object	Yes	Specifies the tested port. Syntax: `{ "value": "boolean", "tag": "integer" }`
is_port_open	object	No	Specifies whether the tested port is open. Syntax: `{ "value": "boolean", "tag": "integer" }`
hostname	object	Yes	The hostname tested (can be an IP address).
reverse_dns	object	Yes	The reverse DNS for the IP tested.
http_response	object	No	The HTTP response code to a GET request. Absent if server does not support SSL/TLS.
server_signature	object	No	The content of Server HTTP Header. Absent if server does not support SSL/TLS.
protocol	object	Yes	Specifies the used protocol (e.g. HTTPS).

3. Certificates

The certificates section includes certificate information, chains and graphs. It is composed of 4 main subparts. The subparts will be detailed later in the document:

information: a list containing detailed information about server certificates.

chain_installation_issues: list of certificate chain installation issues.

chains: a list containing ordered certificates trust paths.

graphs: a list containing the information needed to build graphs.

3.1. Information

As mentioned before, the information part is a list containing all server certificates. One server certificate has the following attributes:

Field Name	Type	Always present	Description
key_type	string	Yes	Contains the type of key associated (RSA, ECDSA…).
key_size	integer	Yes	Contains the size of the key associated in bits.
signature_ algorithm	string	Yes	Contains the signature algorithm used to sign the certificate.
cn	string	Yes	Contains Common Name of the certificate.
issuer_cn	string	Yes	The common name of the issuer.
o	string	Yes	Contains the name of the organization.
san	string	Yes	Contains the Subject Alternative Names for which the certificate is valid.
transparency	bool	Yes	Set to true if the certificate provides transparency.
ev	bool	Yes	Set to true if the certificate provides Extended Validation.
validation	string	Yes	Contains details on validation.

valid_from	integer	Yes	The date from which the certificate is valid.
valid_to	integer	Yes	The expiration date of the certificate.
valid_now	bool	Yes	Set to true if the certificate is valid at the time of testing.
expires_soon	bool	Yes	Set to true if the certificate expires in less than 30 days.
ocsp_must_ staple	bool	Yes	Set to true if the certificate has must staple extension.
supports_ocsp_ stapling	bool	Yes	Set to true if the certificate supports OCSP stapling.
self_signed	bool	Yes	Set to true if the certificate is self-signed.
valid_for_host	bool	Yes	Set to true if the certificate is valid for the domain tested.
valid_for_ptr	bool	Yes	Set to true if the certificate is valid for the domain that can be found by reverse DNS search of the IP address.
skipped	bool	Yes	Set to “true” if an IP address has been tested, and the server sent a valid certificate for the hostname from the PTR record.
revoked	bool	Yes	Set to true if the certificate has been revoked.
known_issuer	bool	Yes	Set to true if the CA that signed the certificate is trusted.
trusted	bool	Yes	Set to true if the certificate can be trusted.
revocaction_ information	object	Yes	This object contains information about revocation.

An example of “revocation_information” object:

```
"revocation_information": {
          "ocsp": {
            "url": "http://ocsp.comodoca.com",
            "revoked": false,
            "error": false
          },
          "crl": {
            "url": "http://crl.comodoca.com/cPanelIncCertificationAuthority.crl",
            "revoked": false,
            "error": false
          }
}
```

3.2. Chain Installation Issues

This section contains a list of several installation issue checkups for server certificates. There's a boolean field '**value**' which is set to '**True**' if there is any issue detected, along with an array '**results**' annotating the results of separate checks:

```
    "chain_installation_issues": [
      {
        "value": false,
        "results": {…}
      }
    ]
```

The structure of the '**results**' object is as follows:

Field Name	Type	Always present	Description
is_chain_complete	obj	Yes	Set to true if server sends intermediate certs for at least one chain.
has_sent_root_ca	obj	Yes	Set to true if the server sends root CA in the cert chain.
is_order_correct	obj	Yes	Set to true if certificate chain was provided in correct order.
has_sent_extra_ certs	obj	Yes	Set to true if server has sent certs that were not expected.
chain_rely_on_ expired_cert	obj	Yes	Set to true if the chain relies on an expired certificate.

The syntax of all aforementioned objects is as follows:

```
{
"value": "boolean", "message_id": "integer",
  "tag": "integer", "message": "string"
}
```

Where “message_id” value corresponds with the id listed in the Appendix 1 of this document.

3.3. Chains

The chain section contains a list of certificate chains that have been reconstructed from the server certificates. A certificate chain is an ordered list of certificates from the server certificate (leaf certificate) to the root CA certificate.

The structure is as follows:

Field Name	Type	Always present	Description
data_pem	string	Yes	Contains the PEM data.
sha256	string	Yes	Contains the sha256 sum of the certificate.
cn	string	Yes	Contains Common Name of the certificate.
key_type	string	Yes	Contains the type of key associated (RSA, CDSA…).
key_size	integer	Yes	Contains the size of the key associated in bits.
signature_ algorithm	string	Yes	Contains the signature algorithm used to sign the certificate.
valid_to	integer	Yes	The expiration date of the certificate.
valid_from	integer	Yes	The date from which the certificate is valid.
pin	string	Yes	The pin of the corresponding public key, used in HPKP.
matches_hpkp	bool	Yes	Set to true if this certificate is pinned.
cert_type	string	Yes	Contains one of the following cert types: • Root CA • Intermediate CA • Server certificate
comment	string	Yes	Contains one of the following comments: • Self-signed • Extended Validation
weak_key_size	bool	Yes	Set to true if key size is small for specific signature algorithm.
weak_signature_ algorithm	bool	Yes	Set to true if signature algorithm is weak.

3.4. Graphs

The graphs section contains mainly the same information as the chains part, but with level and children information and with mostly removed duplicates from chains. This way, it is possible to draw relationships between certificates. The structure is as follows:

Field Name	Type	Always present	Description
data_pem	string	Yes	Contains the PEM data.
sha256	string	Yes	Contains the sha256 sum of the certificate.
cn	string	Yes	Contains Common Name of the certificate.
key_type	string	Yes	Contains the type of key associated (RSA, CDSA…).
key_size	integer	Yes	Contains the size of the key associated in bits.
signature_ algorithm	string	Yes	Contains the signature algorithm used to sign the certificate.
valid_to	integer	Yes	The expiration date of the certificate.
valid_from	integer	Yes	The date from which the certificate is valid.
pin	string	Yes	The pin of the corresponding public key, used in HPKP
matches_hpkp	bool	Yes	Set to true if this certificate is pinned.
cert_type	string	Yes	Contains one of the following cert types: • Root CA • Intermediate CA • Server certificate
comment	string	Yes	Contains one of the following comments: • Self-signed • Extended Validation
weak_key_size	bool	Yes	Set to true if key size is small for specific signature algorithm
weak_signature_ algorithm	bool	Yes	Set to true if signature algorithm is weak.
children_hashes	array	Yes	Every string contained is the sha256 sum of certificates that have been signed with the current one.
tree_levels	array	Yes	Every integer contained in this array corresponds to the level of the certificate in the chain, starting from the server certificate. Server certificates have level=0.

4. NIST and HIPAA

NIST and HIPAA sections contain all information related to NIST and HIPAA compliance respectively. NIST and HIPAA sections have the same structure and syntax.

Please note: most of the objects in this list have the following syntax, unless specified otherwise. None of them is present if the server does not support SSL/TLS. Syntax:

```
{       "value": "boolean",
        "message_id": "integer",
        "tag": "integer",
        "description_id": "integer",
        "title_id": "integer",
        "visible": "boolean"
}
```

Where message_id, tag, description_id and title_id correspond with their respective counterparts in the appendix section of this document.

Field Name	Type	Always present	Description
compliant	bool	No	Set to true if server is compliant with NIST / HIPAA guidelines. Syntax: `{ "value": "bool" }`
cert_x509_v3	object	No	Set to true if the server certificate is an X509 certificate in version 3.
cert_self_signed	object	No	Set to true if the server cert is self-signed.
cert_provides_ revocation_ information	object	No	Set to true if the certificate provides revocation information.
cert_small_key	object	No	Set to true if the private key is too small.
cert_signature_ algorithm_ mismatch	object	No	Set to true if the cert has been signed with a wrong algorithm.
cert_weak_ signature	object	No	Set to true if the cert has not been signed using SHA2
supports_invalid_ protocols	object	No	Set to true if the server supports protocols that are not approved by NIST / HIPAA.
supports_invalid_ cipher_suites	object	No	Set to true if the server supports cipher suites that are not approved by NIST / HIPAA.

dh_parameter_ weak	object	No	Set to true if the Diffie-Hellman parameter size is below NIST requirements (2048).
dh_parameter_ size	object	No	Gives the size of the Diffie-Hellman parameter in bits. Syntax: `{ "value": "integer", "message_id": "integer", "tag": "integer", "description_id": "integer", "title_id": "integer", "visible": "boolean" }`
supports_invalid_ curves	object	No	Set to true if the server supports elliptic curves that are not approved by NIST / HIPAA.
supports_ mandatory_ curves	object	No	Set to true if the server supports at least one of the mandatory curves.
supports_tlsv1.1	object	No	Set to true if the server supports TLSv1.1
supports_tlsv1.2	object	No	Set to true if the server supports TLSv1.2 (only for HIPAA compliance).
supports_tlsv1.3	object	No	Set to true if the server supports TLSv1.3 (only for HIPAA compliance).
supports_ocsp_ stapling	object	No	Set to true if the server supports OCSP stapling.
provides_reneg_ information	object	No	Set to true if server say if it supports or not secure renegotiation.
ec_point_format	object	No	Mixed syntax. Set to true if the server supports ec point format TLS extension. Set to false if not and “not_present” if server doesn’t send TLS Extension.
has_all_ mandatory_ ciphers	object	No	Set to false if the server is missing mandatory ciphers.
missing_ mandatory_ ciphers	array	No	Lists missing mandatory ciphers. Syntax: `"missing_mandatory_ciphers": [ { "value": "string", "tag": "integer",`

			<pre>"protocols": ["string", "string"]
			}</pre>

supported_ elliptic_curves	array	No	Lists supported elliptic curves. Syntax:
			<pre>"supported_elliptic_curves": [
			{ "value": "string",
			"size": "integer",
			"tag": "integer" }
			]</pre>

supported_ protocols	array	No	Lists supported protocols. Only supported versions are present. Syntax:
			<pre>"supported_protocols": [
			{ "value": "string", "tag": "integer" },
			{ "value": "string", "tag": "integer" }
			]</pre>

supported_ cipher_suites	array	No	An array of object contains supported cipher suites. Syntax:
			<pre>"supported_cipher_suites": [
			{
			"value": "string",
			"tag": "integer",
			"protocols": ["string", "string"]
			}
			]</pre>

5. PCI DSS

The following section contains all information related to the PCI DSS compliance.

Please note: most of the objects in this list have the following syntax, unless specified otherwise. None of them is present if the server does not support SSL/TLS. Syntax:

```
{      "value": "boolean",
       "message_id": "integer",
       "tag": "integer",
       "description_id": "integer",
       "title_id": "integer",
       "visible": "boolean"
}
```

Where message_id, tag, description_id and title_id correspond with their respective counterparts in the appendix section of this document.

Field Name	Type	Always present	Description
compliant	bool	No	Set to true if server is compliant with PCI DSS guidelines. Syntax: `{ "value": "bool" }`
cert_small_key	object	No	Set to true if the private key is too small.
cert_weak_ signature	object	No	Set to true if the cert has not been signed using SHA2
cert_trusted	object	No	Set to true if the cert could be trusted.
supported_ protocols	array	No	Lists supported protocols. Only supported versions are present. Syntax: `"supported_protocols": [ { "value": "string", "tag": "integer" }, { "value": "string", "tag": "integer" } ]`
supports_invalid_ protocols	object	No	Set to true if the server supports protocols that are not approved by PCI DSS.
supports_invalid_ cipher_suites	object	No	Set to true if the server supports cipher suites that are not approved by PCI DSS.

supports_invalid_ curves	object	No	Set to true if the server supports elliptic curves that are not approved by PCI DSS.
supports_ insecure_ reneg	object	No	Set to true if the server supports client-initiated insecure renegotiation.
poodle_tls	object	No	Set to true if the server is vulnerable to poodle over TSL.
poodle_ssl	object	No	Set to true if the server is vulnerable to poodle over SSL.
goldendoodle	object	No	Set to true if the server is vulnerable to poodle over GOLDENDOODLE.
zombie_poodle	object	No	Set to true if the server is vulnerable to poodle over Zombie POODLE.
sleeping_poodle	object	No	Set to true if the server is vulnerable to poodle over Sleeping POODLE.
cve_2016_2107	object	No	Set to true if the server is vulnerable to CVE-2016-2107.
cve_2014_0224	object	No	Set to true if the server is vulnerable to CVE-2014-0224.
heartbleed	object	No	Set to true if the server is vulnerable to heartbleed.
drown	object	No	Set to true if the server is vulnerable to drown.
robot	object	No	Set to true if the server is vulnerable to ROBOT.
0length_openssl	object	No	Set to true if the server is vulnerable to poodle over 0-Length OpenSSL.
dh_parameter_ weak	object	No	Set to true if the Diffie-Hellman parameter size is below NIST requirements (2048).
dh_parameter_ size	object	No	Gives the size of the Diffie-Hellman parameter in bits. Syntax: `{ "value": "integer", "message_id": "integer", "tag": "integer", "description_id": "integer", "title_id": "integer", "visible": "boolean" }`
supported_ protocols	array	No	Lists supported protocols. Only supported versions are present. Syntax:

			<pre>"supported_protocols": [
			  { "value": "string", "tag": "integer" },
			  { "value": "string", "tag": "integer" }
			]</pre>

supported_cipher_ suites	array	No	An array of object contains supported cipher suites. Syntax:
			<pre>"supported_cipher_suites": [
			  {
			  "value": "string",
			  "tag": "integer",
			  "protocols": ["string", "string"]
			  }
			]</pre>

supported_ elliptic_ curves	array	No	Lists supported elliptic curves. Syntax:
			<pre>"supported_elliptic_curves": [
			  { "value": "string",
			  "size": "integer",
			  "tag": "integer" }
			]</pre>

6. Industry Best Practices

This section contains information on industry's best practices.

Please note: most of the objects in this list have the following syntax, unless specified otherwise. None of them is present if the server does not support SSL/TLS. Syntax:

```
{
      "value": "boolean",
      "message_id": "integer",
      "tag": "integer",
      "description_id": "integer",
      "title_id": "integer",
      "visible": "boolean",
      "message": "string",
      "title": "string"
}
```

Where message_id, tag, description_id and title_id correspond with their respective counterparts in the appendix section of this document.

Field Name	Type	Always present	Description
cert_valid_too_long	object	No	Set to true if the cert has been signed for more than 3 years.
cert_ev	object	No	Set to true if the cert provides with Extended Validation.
http_to_https_redirect	object	No	Set to true if the server redirects from HTTP to HTTPS.
https_to_http_redirect	object	No	Set to true if the server redirects from HTTPS to HTTP.
mixed_content	object	No	Set to true if HTTP content is included into HTTPS.
supports_tlsv1.3	object	No	Set to true if the server supports TLSv1.3
supports_fallback_scsv	object	No	Set to true if the server supports TLS Fallback SCSV.
supports_client_initiated_reneg	object	No	Set to true if the server supports client-initiated renegotiation.

supports_secure_reneg	object	No	Set to true if the server supports secure renegotiation.
tls_compression	object	No	Set to true if the server supports TLS compression.
has_hsts	object	No	Set to true if the server enforces HSTS.
hsts_duration	object	No	This is the duration of HSTS max-age in seconds: `{ "value": "integer" }`
hsts_long	object	No	Set to true if the server's HSTS max-age is above 180 days.
has_hpkp	object	No	Set to true if the server has HPKP header.
hpkp_valid	object	No	Set to false if HPKP syntax is invalid or if no pin match
hpkp_duration	object	No	This is the duration of HPKP max-age in seconds: `{ "value": "integer" }`
hpkp_long	object	No	Set to true if the server's HPKP max-age is above 60 days
early_data	object	No	Set to true if the server supports 0-RTT early data
has_preference	object	No	Set to true if the server cipher suite preference is enabled.
prefers_weak_ciphers	object	No	Set to true if the server prefers cipher suites that have not been approved by PCI DSS.
prefers_pfs	object	No	Set to true If the server prefers cipher suites providing Perfect Forward Secrecy.
chain_rely_on_expired_cert	object	No	Set to true if the chain relies on an expired certificate
cipher_preference	array	No	Lists cipher suites preferred per protocol. Syntax: `"cipher_preference": [ { "protocol": "string", "value": "string", "tag": "integer" } ]`

7. Email

This section lists SPF, DMARC and DKIM server security hardenings. This section is present only if tested host is email service.

SPF – is a simple email-validation system designed to detect email spoofing by providing a mechanism to allow receiving mail exchangers to check that incoming email from a domain comes from a host authorized by that domain's administrators.

DMARC – is an email-validation system designed to detect and prevent email spoofing.

DKIM – Domain Keys Identified Mail (DKIM) provides email authentication and helps prevent potentially malicious emails from reaching recipients by using a digital signature in the email header. This validates that the email originated from the correct location and was not tampered with in transit. This also protects the reputation of the supposed sender of the email.

Example:


```
{
  "values": [
    {
      "raw": "string",
      "messages": []
    }
  ],
  "highlights": [
    {
      "highlight": "string",
      "highllight_id": "int",
      "tag": "int"
    }
  ],
  "title": "string",
  "title_id": "integer",
  "description": "string",
  "description_id": "int",
  "visible": "boolean"
}
```


8. Third-Party Content

This section contains information concerning third-party content. The structure is as follows:

Field Name	Type	Always present	Description
url	string	Yes	The URL of the third-party content.
method	string	Yes	Specifies the HTTP method used.
redirect_chain	array	Yes	Details the redirect chain.
resource_type	string	Yes	Indicates the resource type.
content_type	string	Yes	Indicates the content type.
status	integer	Yes	Indicates the response status code.
status_text	string	Yes	Details the response status
size	integer	Yes	Indicates the size of the third-party content
md5	string	Yes	Indicates the md5 hash
sha256	string	Yes	Indicates the sha256 hash
selector	string	Yes	Indicates the selector
server	object	Yes	Contains information about the IP and port. Syntax: `{ "ip": "string", "port": "integer" }`
security	object	Yes	Contains information about the protocol. Syntax: `{ "protocol": "string" }`
headers	object	Yes	Contains information about the received headers. Syntax: `{ "request": { "referer": "string", "user-agent": "string" }, "response": {} }` The “response” will be detailed later in the document.

8.1. Third-Party Content Headers Response

Field Name	Type	Always present	Description
date	string	Yes	Indicates the date.
content-encoding	string	Yes	Details the content-encoding.
last-modified	string	Yes	Details when it was modified the last time.
alt-svc	string	No	Details the cf-ray header alt-svc header.
access-control-allow-origin	string	No	Details the access-control-allow-origin header.
age	string	No	Details the age of the content.
cf-cache-status	string	No	Details the cf-cache-status header.
cf-ray	string	No	Details the cf-ray header.
surrogate-key	string	No	Details the surrogate-key.
etag	string	No	Details the ETag header.
server	string	No	Contains details on the server.
vary	string	No	Details the vary header.
strict-transport-security	string	No	Details the strict-transport-security header.
x-cache	string	No	Details the x-cache header.
x-content-type-options	string	No	Details the x-content-type-options header.
x-xss-protection	string	No	Details the x-xss-protection header.
x-ton-expected-size	string	No	Details the x-ton-expected-size header.
expect-ct	string	No	Details on the expect-ct header.
content-type	string	No	Details the content type.
status	string	No	Details the response status code.
cache-control	string	No	Details the cache-control header.
x-hello-human	string	No	Details the x-hello-human header.
accept-ranges	string	No	Details the accept-ranges header.

timing-allow-origin	string	No	Details the timing-allow-origin header.
content-length	string	No	Details the content-length header.
link	string	No	Contains the link.
expires	string	No	Indicates the expiration date.
served-in-seconds	string	No	Details the served-in-seconds header.

9. Internals

Contains internal information such as city, country, server IP and more.

Field Name	Type	Always present	Description
id	string	Yes	The id of the test.
short_id	string	Yes	The short id of the test.
grade_ norm	string	Yes	The grade of the test.
title	string	Yes	The title of the test.
heading	string	Yes	The heading of the test.
server_ip	string	Yes	The IP address of the tested server.
city	string	Yes	The city of the tested server.
country	string	Yes	The country of the tested server.
title_twitter	string	Yes	The official title of the test to be displayed on twitter.
description	string	Yes	A description of the free service.
description_twitter	string	Yes	The description of the free service to be displayed on twitter.
can_ index	bool	Yes	Set to 'true' if the result can be indexed.
errors	integer	Yes	Contains the number of errors.
scores	object	Yes	Contains information about the number of found issues concerning NIST, HIPAA and PCI DSS.

10. Highlights

This part lists highlights in an ordered way. The syntax is the following:

```
[{ "highlight_id": "integer", "tag": "integer", "highlight": "string" }]
```

Just like “message_id” or “description_id”, the “highlight_id” contains the index to the corresponding text in appendix 4 of this document.

11. Results

Results part contains the main results of the test. The structure is as follows:

Field Name	Type	Always present	Description
has_ssl_tls	bool	Yes	Set to “true” if the server supports SSL/TLS.
score	integer	Yes	Server’s score.
grade	string	Yes	Server’s grade.
is_blacklisted	bool	Yes	Set to true if email server is blacklisted.

Appendix 1: List of Tag values

Below is a reminder of the list of tags that are used:

Tag value	Description
0	Nothing, empty
1	Good configuration
2	Not compliant with NIST guidelines
3	Misconfiguration or weakness
4	Information
5	Non-compliant with PCI DSS requirements
6	Not compliant with NIST and PCI DSS
7	Not vulnerable
8	Deprecated. Dropped in June 2018
9	Non-compliant with HIPAA guidance
10	Non-compliant with NIST and HIPAA
11	Non-compliant with HIPAA and PCI DSS
12	Non-compliant with NIST, HIPAA and PCI DSS
13	No Encryption

Appendix 2: List of Message values

ID	Value
1	The version of the RSA X509 certificate provided by the server is prior to version 3 (the latest one).
2	The version of the ECDSA X509 certificate provided by the server is prior to version 3 (the latest one).
3	The version of the following X509 certificates provided by the server is prior to version 3 (the latest one): RSA, ECDSA.
4	Some of the X509 certificates provided by the server are prior to version 3 (the latest one).
5	All the X509 certificates provided by the server are in version 3.
6	The RSA certificate provided by the server is self-signed.
7	The ECDSA certificate provided by the server is self-signed.
8	The following certificates are self-signed: RSA, ECDSA.
9	Some of the certificates provided by the server are self-signed.
10	All the certificates provided by the server have been signed by a CA.
11	The RSA certificate provided is missing OCSP URI and crlDistributionPoints extension, making impossible to verify if it has been revoked.
12	The ECDSA certificate provided is missing OCSP URI and crlDistributionPoints extension, making impossible to verify if it has been revoked.
13	The following certificates are missing OCSP URI and crlDistributionPoints extension, making impossible to verify if they have been revoked: RSA, ECDSA.
14	Some of the certificates provided are missing OCSP URI and crlDistributionPoints extension, making impossible to verify if they have been revoked.
15	All the certificates sent by the server provide ways to check their revocation status.
16	The RSA certificate's key length is too small.
17	The ECDSA certificate's key length is too small.
18	The following certificates' key lengths are too small: RSA, ECDSA.
19	Some of the certificates have a public key that is too small.
20	All the certificates provided have public keys that are long enough.
21	The RSA certificate provided has not been signed using the proper algorithm according to NIST guidelines.

22	The ECDSA certificate provided has not been signed using the proper algorithm according to NIST guidelines.
23	The following certificates have not been signed using the proper algorithm according to NIST guidelines: RSA, ECDSA.
24	Some of the certificates provided have not been signed using the proper algorithm according to NIST guidelines.
25	All the certificates provided have been signed using the proper algorithm.
26	The RSA certificate provided has been signed using a weak algorithm.
27	The ECDSA certificate provided has been signed using a weak algorithm.
28	The following certificates have been signed using a weak algorithm: RSA, ECDSA.
29	Some of the certificates provided have been signed using a weak algorithm.
30	All the certificates provided have been signed using a strong algorithm.
31	The RSA certificate provided has been validated for more than 3 years. This means that the private key of the server will remain the same for more than 3 years. NIST guidelines suggest limiting certificate validity to 3 years maximum.
32	The ECDSA certificate provided has been validated for more than 3 years. This means that the private key of the server will remain the same for more than 3 years. NIST guidelines suggest limiting certificate validity to 3 years maximum.
33	The following certificates have been validated for more than 3 years: RSA, ECDSA. This means that the private keys of the server will remain the same for more than 3 years. NIST guidelines suggest limiting certificate validity to 3 years maximum.
34	Some of the certificates provided have been validated for more than 3 years. This means that the private keys of the server will remain the same for more than 3 years. NIST guidelines suggest limiting certificate validity to 3 years maximum.
35	All the certificates provided have been validated for less than 3 years.
36	The RSA certificate provided by the server could not be trusted.
37	The ECDSA certificate provided by the server could not be trusted.
38	The following certificates provided by the server could not be trusted: RSA, ECDSA.
39	Some of the certificates provided by the server could not be trusted.
40	All the certificates provided by the server are trusted.
41	The RSA certificate provided is NOT an Extended Validation (EV) certificate.
42	The ECDSA certificate provided is NOT an Extended Validation (EV) certificate.
43	The following certificates are NOT Extended Validation (EV) certificates: RSA, ECDSA.
44	Some of the certificates provided are NOT Extended Validation (EV) certificates.
45	All the certificates provided by the server are Extended Validation (EV) certificates.

46	The HTTP version of the website does not redirect to the HTTPS version. We advise to enable redirection.
47	The HTTP version of the website redirects to the HTTPS version.
48	The website includes HTTP content in HTTPS.
49	The Diffie-Hellman parameter's size is only $value bits. A longer one must be generated to prevent Logjam vulnerability.
50	The server does not support P-256 or P-384 curves which are required by NIST guidelines.
51	The support of TLSv1.2 is required minimum according to NIST guidelines.
52	The server supports TLSv1.1. It's still compliant, but NIST recommends to drop TLS 1.1 support since SP 800-52 REV. 2
53	The server supports TLSv1.3 which is the only version of TLS that currently has no known flaws or exploitable weaknesses.
54	The server does not support TLSv1.3 which is the only version of TLS that currently has no known flaws or exploitable weaknesses.
55	The server does not prefer cipher suites. We advise to enable this feature in order to enforce usage of the best cipher suites selected.
56	The server enforces cipher suites preference.
57	The server prefers cipher suite that has not been approved by PCI DSS requirements for at least one of the supported protocols.
58	For TLS family of protocols, the server prefers cipher suite(s) providing Perfect Forward Secrecy (PFS).
59	The server does not prefer cipher suites providing strong Perfect Forward Secrecy (PFS). We advise to configure your server to prefer cipher suites with ECDHE or DHE key exchange.
60	The server provides HTTP Strict Transport Security for more than 6 months: $value seconds
61	The server provides HTTP Strict Transport Security for less than 6 months: $value seconds
62	The server does not enforce HTTP Strict Transport Security. We advise to enable it to enforce the user to browse the website in HTTPS.
63	The server provides HPKP for more than 2 months: $value seconds
64	The server provides HPKP for less than 2 months: $value seconds
65	The server sends an invalid HPKP header: the certificate chain does not match the signature sent, or the syntax is invalid. We advise to review your configuration.

66	The server does not enforce HTTP Public Key Pinning that helps preventing man-in-the-middle attacks.
67	The server supports TLS_FALLBACK_SCSV extension for protocol downgrade attack prevention.
68	TLS_FALLBACK_SCSV extension prevents protocol downgrade attacks. We advise to update your TLS engine to support it.
69	The server is vulnerable to POODLE over TLS.
70	The server's response to invalid TLS packet is not compliant with RFC 4346 (section 6.2.3.2) and may be an indicator that the server is vulnerable to POODLE over TLS.
71	The server is not vulnerable to POODLE over TLS.
72	The server is vulnerable to OpenSSL padding-oracle flaw (CVE-2016-2107).
73	The server is not vulnerable to OpenSSL padding-oracle flaw (CVE-2016-2107).
74	The server may be vulnerable to OpenSSL padding-oracle flaw (CVE-2016-2107), make sure that your OpenSSL version is up to date.
75	The server supports a client-initiated secure renegotiation that may be unsafe and allow Denial of Service attacks.
76	The server does not support client-initiated secure renegotiation.
77	The server supports a client-initiated insecure renegotiation that is unsafe and may allow Man-In-The-Middle attacks.
78	The server does not support client-initiated insecure renegotiation.
79	The server supports OCSP stapling, which allows better verification of the certificate validation status.
80	The server does not support OCSP stapling. Its support allows better verificationof the certificate validation status.
81	The server supports secure server-initiated renegotiation.
82	The server does not support secure server-initiated renegotiation.
83	The server does not provide information if the client should accept secure server-initiated renegotiation requests.
84	TLS compression is supported by the server which may allow CRIME attack. We advise to disable this feature.
85	TLS compression is not supported by the server.
86	The server supports elliptic curves but not the EC_POINT_FORMAT TLS extension.
87	The server supports the EC_POINT_FORMAT TLS extension.
88	The server version of OpenSSL is vulnerable to Heartbleed attack allowing remote compromise of your server. Update your OpenSSL to the latest version urgently!

89	The server version of OpenSSL is not vulnerable to Heartbleed attack.
90	The server is vulnerable to CVE-2014-0224 (OpenSSL CCS flaw).
91	The server may be vulnerable to CVE-2014-0224 (OpenSSL CCS flaw), make sure that your OpenSSL version is up to date.
92	The server is not vulnerable to CVE-2014-0224 (OpenSSL CCS flaw).
93	Diffie-Hellman parameter size: $value bits
94	The server is not vulnerable to the DROWN attack.
95	The server is vulnerable to the DROWN attack. SSLv2 must be disabled urgently!
96	The server is not vulnerable to POODLE over SSL.
97	The server is vulnerable to POODLE over SSL. SSLv3 should be disabled.
98	The RSA certificate provided has not been signed using the proper algorithm according to HIPAA guidance.
99	The ECDSA certificate provided has not been signed using the proper algorithm according to HIPAA guidance.
100	The following certificates have not been signed using the proper algorithm according to HIPAA guidance: RSA, ECDSA.
101	Some of the certificates provided have not been signed using the proper algorithm according to HIPAA guidance.
102	The server does not support P-256 or P-384 curves which are required by HIPAA guidance.
103	The support of TLSv1.1 is required minimum according to HIPAA guidance.
104	The server supports TLSv1.1 which is required minimum to comply with HIPAA guidance.
105	Intermediate certificate is provided by the server.
106	Intermediate certificate is not provided by the server.
107	Server sends an unnecessary root certificate.
108	No unnecessary root certificate sent by the server.
109	The chain provided is in correct order.
110	Server provides certificate chain in a wrong order.
111	Server sends useless certificates.
112	Server does not send useless certificates.
113	The server does not support OCSP stapling for its RSA certificate. Its support allows better verification of the certificate validation status.
114	The server does not support OCSP stapling for its ECDSA certificate. Its support allows better verification of the certificate validation status.

115	The server does not support OCSP stapling for its RSA and ECDSA certificates. Its support allows better verification of the certificate validation status.
116	The server does not support OCSP for some of the provided certificates. Its support allows better verification of the certificate validation status.
117	The server supports OCSP stapling, which allows better verification of the certificate validation status.
118	HTTPS version of the website redirects to HTTP. This is a bad practice since visitors are being redirected from a secure version of the site to an insecure one.
119	This domain has a Certification Authority Authorization (CAA) record.
120	This domain does not have a Certification Authority Authorization (CAA) record.
121	The server is vulnerable to ROBOT (Return Of Bleichenbacher's Oracle Threat) vulnerability.
122	The server is not vulnerable to ROBOT (Return Of Bleichenbacher's Oracle Threat) vulnerability.
123	SPF syntax is not valid.
124	There is no 'exp' or 'redirect' domain defined.
125	There is no 'all' or 'a' or 'mx' or 'ptr' or 'ip4' or 'ip6' or 'exist' inside syntax.
126	There is no valid domain name.
127	Multiple 'redirect' modifiers detected, not in line with RFC 7208.
128	Multiple 'exp' modifiers detected, not in line with RFC 7208.
129	Invalid or not set DMARC version.
130	Missing 'p' (Requested handling policy) action.
131	Invalid 'p' (Requested handling policy) action, valid are 'none', 'reject', 'quarantine'.
132	DMARC syntax is not valid.
133	Invalid 'rf' (Failure reporting format(s)) field value, allowed are 'iodef' or 'afrf'.
134	Invalid 'pct' (Sampling rate) field value, allowed is integer in range of 0 up to 100.
135	Invalid 'ri' (Aggregate Reporting interval) field value, allowed is integer in range of 0 up to 4294967295.
136	Invalid 'ruf' (Reporting URI(s) for failure data) field value, valid is mailto:email@domain.com and/or http://domain.com.
137	Invalid 'rua' (Reporting URI(s) for aggregate data) field value, valid is mailto:email@domain.com and/or http://domain.com.
138	Invalid 'fo' (Failure reporting options) field value, valid are 0/1/d/s or [0/1/d/s]:[0/1/d/s].

139	Invalid 'aspf' (SPF alignment mode) field value, valid are 'r' or 's'
140	Invalid 'adkim' (DKIM alignment mode) field value, valid are 'r' or 's'
141	Invalid 'sp' (Requested handling policy for subdomains) field value, valid are 'none', 'reject', 'quarantine'.
142	DKIM syntax is not valid.
143	Invalid DKIM version, the only valid value is 'DKIM1'.
144	Invalid 'k' (Key type), the only valid value is 'rsa'.
145	Invalid 'g' (Granularity of the key), if set it must not be empty or have multiple *.
146	Invalid 'h' (Acceptable hash algorithm(s)), valid values are 'sha1' or 'sha256' or '' (empty) allowing all.
147	Invalid 's' (Service type), valid values are 'email' or '*' or '' (empty) allowing all.
148	Invalid 't' (Flags type), valid values are 'y' or 's' or '' (empty) no flag set.
149	Test mode is on. The 'y' flag tells recipients to ignore your DKIM signature.
150	Syntax error, key #key_name# is not good DKIM key record.
151	Syntax error, key #key_name# is not good DMARC key record.
152	Missing mandatory field, 'p' (Public Key).
153	Public key is not valid.
154	Public key is properly set, with size of #pub_key_size# bits.
155	Public key is smaller than minimum 1024 bits.
156	Public key is >= 4096 bits, it may not fit in DNS UDP query.
157	Expect-CT header is properly set.
158	Expect-CT header is not properly set.
159	The server does not send EC_POINT_FORMAT TLS extension according to RFC 4492 (section 5.2, page 15).
160	The server supports TLSv1.2.
161	The server does not support TLSv1.2.
162	Server's TLSv1.3 Early Data (RFC 8446, page 17) is properly implemented.
163	Server's TLSv1.3 Early Data (RFC 8446, page 17) is not enabled.
164	The server is vulnerable to GOLDENDOODLE.
165	The server is not vulnerable to GOLDENDOODLE.
166	The server is vulnerable to Zombie POODLE.
167	The server is not vulnerable to Zombie POODLE.

168	The server is vulnerable to Sleeping POODLE.
169	The server is not vulnerable to Sleeping POODLE.
170	The server is vulnerable to 0-Length OpenSSL.
171	The server is not vulnerable to 0-Length OpenSSL.
201	Certificate chain rely on expired certificate.
202	Certificate chain does not rely on expired certificate.
204	Certificate chain rely on expired certificate, it can break connection for some clients.

Appendix 3: List of Description values

ID	Value
1	For compatibility reasons, NIST requires the server to provide X509 certificates inversion 3.
2	The trust model of PKI certificates currently resides on the fact they are signed by known Certificate Authority (CA), or a CA that we choose to trust. Self-signed certificates cannot be trusted.
3	PKI certificate contains the server's public key, enabling users to encrypt messages sent to server that on its side will decrypt them using its private key. In case of the loss or compromise of the server's private key, the certificate cannot be trusted anymore and must be revoked and markes as untrusted. However, if a certificate does not contain revocation information, it is impossible to check if it has been revoked or not.
4	Asymmetric cryptography uses a public key to encrypt messages, or verify, signatures and a private key to decrypt or sign messages. If the key size is too short there is a risk that an attacker can forge the private key and potentially decrypt all traffic between the client and the server.
5	To be trusted, a certificate is hashed using a specific algorithm in order to get a statistically unique fingerprint to sign it. However, the fingerprint is not mathematically unique and an attacker may forge false certificate with the samehash value to impersonate the server if the hash algorithm used to sign it is too weak.

6	NIST guidelines specify that certificate should not be signed for more than 3 years. In general it is a good practice to renew private key of the server every 1 to 3 years, in order to prevent attacker forging it from the public key.
7	In order to be trusted, a certificate must be signed by a trusted Certificate Authority (CA), the DNS name of the server must match either the Common Name of the certificate or its Subject Alternative Names, it must be valid at the current date (not expired) and it must not have been revoked.
8	Redirecting the users from the HTTP to the HTTPS version is a good practice to enforce secure browsing.
9	When the HTTPS version of a website contains insecure elements, it cannot be totally trusted. Attackers can still intercept these elements which can contain personnal data, or tamper with them to include malicious content in.
10	It is a common best practice to configure TLS servers to have a cipher suite preference, in order to enforce the best compromise between security and performance.
11	Enforcing server preference needs to carefully order supported cipher suites. Preferring a weak cipher suite will cause every browser supporting it to use it instead of a secure one.
12	Perfect-Forward-Secrecy (PFS), based on Diffie-Hellman Ephemeral key exchange, improves global security of TLS. With RSA, an attacker can intercept encrypted communications and record them in order to decrypt them later if he manages to obtain one private key. However, with PFS, it is not possible to use the private key to decrypt messages intercepted in the past.
13	HTTP-Strict-Transport-Security directs a server to force a user's browser to make all subsequent requests via HTTPS for a specified duration.
14	Public-Key-Pinning allows a server to direct a user's browser to remember a list of trusted certificate signatures for a specified duration. These can either be server or CA certificates.
15	When using CBC cipher suites, TLS imposes padding to be filled with its own length. SSLv3 allows padding of any size, which could allow a POODLE attack. POODLE over TLS is a vulnerability that appears when a server does not check the padding value when using CBC cipher suites.
16	OpenSSL padding-oracle flaw (CVE-2016-2107, CVSSv3 5.9/10) has been introduced because of an incorrect fix for the Lucky13 vulnerability and allows attacker to reveal

	encrypted data. It only affects servers supporting hardware acceleration for AES encryption.
17	Client-initiated secure renegotiation (CVE-2011-1473, CVSSv2: 5.0/10) is a vulnerability that may allow Denial of Service (DoS) attacks on servers supporting it.
18	Client-initiated insecure renegotiation (CVE-2009-3555, CVSSv2: 5.8/10) is a vulnerability that may allow an attacker to successfully perform Man-in-The-Middle attacks.
19	Heartbleed (CVE-2014-0160, CVSSv2: 5.0/10) is an OpenSSL vulnerability allowing attackers to access random portions of data stored in the server's memory. It could include user or admin passwords, private keys and other sensitive data.
20	OpenSSL Change-Cipher-Specs flaw (CVE-2014-0224, CVSSv2: 5.8/10) is a vulnerability affecting OpenSSL and allowing an attacker to perform Man-in-The-Middle attacks to downgrade the cipher suite in use between client and server.
21	DROWN vulnerabilty (CVE-2016-0800, CVSSv3 5.9/10) allows attackers to send specially crafted SSLv2 transactions to decrypt TLS connections on servers that use the same RSA private key.
22	POODLE vulnerability (CVE-2014-3566, CVSSv2 4.3/10) is a flaw present in the definition of the SSLv3 protocol. It may allow attackers to decrypt traffic between a browser and a server that use SSLv3 with cipher suties using CBC operation mode.
23	For compatibility reasons, HIPAA guidance requires the server to provide X509 certificates in version 3.
24	Redirecting the users from HTTPS to HTTP is a major security risk.
25	The CAA record specifies which certificate authorities are allowed to issue certificates for the domain in question.
26	ROBOT permits to decrypt intercepted TLS traffic, if the session key is encrypted with RSA algorithm and padding system is PKCS #1 1.5, by a new exploitation technique of a vulnerability discovered in 1998 by Daniel Bleichenbacher.
27	SPF is a simple email-validation system designed to detect email spoofing by providing a mechanism to allow receiving mail exchangers to check that incoming mail from a domain comes from a host authorized by that domain's administrators.
28	DMARC is an email-validation system designed to detect and prevent email spoofing.

29	DomainKeys Identified Mail (DKIM) provides email authentication and helps prevent potentially malicious emails from reaching recipients by using a digital signature in the email header. This validates that the email originated from the correct location and was not tampered with in transit. This also protects the reputation of the supposed sender of the email.
30	Expect-CT allows a site to determine if they are ready for the upcoming Chrome requirements and/or enforce their CT policy.
31	Proper implementation of the Early Data allows a client to use zero round trip (0-RTT) and mitigates some vectors of the Replay Attack.
32	GOLDENDOODLE can be used to hijack authenticated TLS sessions if the server reveals the padding validity of application data records in such a way that a MiTM attacker can recognize well-formed padding independently form a valid Message Authentication Code (MAC). This includes, but is not limited to, cases such as Cisco ASA CVE-2015-4458 where systems completely fail to validate MAC.
33	POODLE TLS and Zombie POODLE both exploit server stacks which behave differently when receiving TLS records with valid MAC and invalid (non-deterministic) padding.
34	Sleeping POODLE exploit server stacks with invalid padding with valid MAC.
35	0-Length OpenSSL exploit server stacks with invalid MAC and 0-length record or valid padding and 0-length record.
36	The server does not respond to every OCSP request. Try to debug this with '-status' parameter of 'openssl' command.

Appendix 4: List of Highlights values

ID	Value
3	The server's certificate is untrusted.
4	The server's Diffie-Hellman parameter is too small, its size is only is %d bits.
6	The TLS engine does not support a TLS version newer than TLSv1.0 and is outdated.
7	The server supports encryption protocols that are insecure and have known security flaws or weaknesses.
8	The server supports cipher suites that are not approved by PCI DSS requirements, HIPAA guidance and NIST guidelines.
9	The server supports cipher suites that are not approved by HIPAA guidance and NIST guidelines.
10	The server prefers cipher suites supporting Perfect-Forward-Secrecy.

13	The server sends an invalid HPKP header.
15	The server is vulnerable to POODLE over TLS.
16	The server is vulnerable to OpenSSL padding-oracle flaw (CVE-2016-2107).
18	The server is vulnerable to Heartbleed.
19	The server is vulnerable to CVE-2014-0224 (OpenSSL CCS flaw).
20	The server seems to require certificate-based authentication.
21	The server configuration seems to be good, but is not compliant with PCI DSS requirements, HIPAA guidance and NIST guidelines.
22	The server configuration seems to be good, but is not compliant with HIPAA guidance and NIST guidelines.
23	The server configuration seems to be good, but is not compliant with PCI DSS requirements.
24	The server is vulnerable to the DROWN attack.
25	The server is vulnerable to POODLE over SSL.
26	Server supports HTTPS but it is configured to redirect to HTTP. This is a major security and privacy risk.
27	The HTTPS port (%d) is closed, data exchange with the remote web server can be intercepted.
28	The server does not tolerate certain TLS versions. This may be a sign of improperTLS implementation.
29	The server configuration supports only TLSv1.2 protocol, precluding users with older browsers from accessing your website.
30	The server configuration has a good protocol compatibility, allowing users with older browsers to access your website.
31	The certificate's CA is not trusted by modern browsers.
32	Test results are over one-week-old, click "Refresh" to update the results.
33	The server is vulnerable to ROBOT (Return Of Bleichenbacher's Oracle Threat) vulnerability.
34	NIST Update to Current Use and Deprecation of TDEA abrogates 3DES authorized in the NIST guidelines.
35	SPF record is missing.
36	SPF record is set.
37	Multiple SPF records found.
38	DMARC record is missing.

39	DMARC record is set.
43	SPF record is set properly.
44	Multiple DMARC records are found.
45	DMARC record is set properly.
50	DKIM record is missing.
51	DKIM record is set.
52	Multiple DKIM records are set.
53	DKIM record is set properly.
54	Email server's SPF, DMARC and DKIM are properly set.
55	Email server's DMARC and DKIM are properly set.
56	Email server's SPF and DKIM are properly set.
57	Email server's SPF and DMARC are properly set.
58	Email server's DKIM is properly set.
59	Email server's DMARC is properly set.
60	Email server's SPF is properly set.
62	CA of your SSL certificate is distrusted by Google Chrome and Mozilla Firefox.
64	Your SSL certificate is distrusted by Google Chrome and Mozilla Firefox.
65	The server supports the most recent and secure TLS protocol version of TLS 1.3.
66	The server configuration supports only TLSv1.2 and TLSv1.3 protocols, precluding users with older browsers from accessing your website.
67	The server configuration supports only TLSv1.3 protocol, precluding users with older browsers from accessing your website.
68	DKIM records are set properly.
70	The server has TLS 1.0 enabled. Since the 30th of June 2018 it is non-compliant with PCI DSS 3.2.1.
71	It seems that your system is blocking one of our IPs 192.175.111.228, 192.175.111.229, 64.15.129.102, 64.15.129.106, 70.38.27.248, 72.55.136.156, 72.55.136.199 please whitelist them for successful continuation of the test.
72	The tested service seems to be a %s.
73	The server's private RSA key is weak.
74	The server's private ECDSA key is weak.
75	The server's private RSA and ECDSA keys are weak.
76	The server's private key is weak.

77	The server's RSA certificate was signed using a weak algorithm.
78	The server's ECDSA certificate was signed using a weak algorithm.
79	The server's RSA and ECDSA certificate were signed using a weak algorithm.
80	The server's certificate is signed using a weak algorithm.
81	The remote server port %d is closed. No SSL/TSL security can be tested.
82	The website is accessible only over unencrypted HTTP protocol
83	Secure connection to the HTTPS port (443) can't be established. No SSL/TSL security can be tested.
84	No certificates were sent by the server.
85	PTR of the IP address of the server matches certificates CN or SAN.
200	Certificate affected by Let's Encrypt CAA problem.
203	Certificate chain rely on expired certificate, it can break connection for some clients.
204	The server supports OCSP stapling, but does not respond to every OCSP request.
700	The server has TLS 1.1 enabled. NIST recommends to drop TLS 1.1 support since SP 800-52 REV. 2

Appendix 5: List of Title values

ID	Value
1	X509 CERTIFICATES ARE NOT IN VERSION 3
2	X509 CERTIFICATES ARE IN VERSION 3
3	CERTIFICATES ARE SELF-SIGNED
4	CERTIFICATES DO NOT PROVIDE REVOCATION INFORMATION
5	CERTIFICATES' KEY ARE WEAK
6	SERVER CERTIFICATES ARE SIGNED WITH A WRONG ALGORITHM
7	CERTIFICATES HAVE A WEAK SIGNATURE
8	CERTIFICATES HAVE BEEN SIGNED FOR MORE THAN 3 YEARS
9	CERTIFICATES ARE UNTRUSTED
10	CERTIFICATES ARE TRUSTED
11	CERTIFICATES DO NOT PROVIDE EV
12	CERTIFICATES PROVIDE EV
13	HTTP SITE DOES NOT REDIRECT

14	ALWAYS-ON SSL
15	MIXED CONTENT
16	DIFFIE-HELLMAN PARAMETER WEAK
17	DIFFIE-HELLMAN PARAMETER SIZE
18	NO SUPPORT FOR COMMON CURVES
19	SERVER DOES NOT SUPPORT TLSv1.1
20	TLSv1.1 SUPPORTED
21	TLSv1.3 SUPPORTED
22	SERVER DOES NOT SUPPORT TLSv1.3
23	SERVER DOES NOT HAVE CIPHER PREFERENCE
24	SERVER HAS CIPHER PREFERENCE
25	SERVER PREFERS WEAK CIPHER SUITES
26	SERVER PREFERS CIPHER SUITES PROVIDING PFS
27	SERVER DOES NOT PREFER CIPHER SUITES PROVIDING PFS
28	SERVER PROVIDES HSTS WITH LONG DURATION
29	SERVER PROVIDES HSTS WITH SHORT DURATION
30	SERVER DOES NOT PROVIDE HSTS
31	SERVER PROVIDES HPKP WITH LONG DURATION
32	SERVER PROVIDES HPKP WITH SHORT DURATION
33	SERVER PROVIDES INVALID HPKP
34	SERVER DOES NOT PROVIDE HPKP
35	TLS_FALLBACK_SCSV
36	POODLE OVER TLS
37	CVE-2016-2107
38	SERVER SUPPORTS CLIENT-INITIATED SECURE RENEGOTIATION
39	SERVER DOES NOT SUPPORT CLIENT-INITIATED SECURE RENEGOTIATION
40	SERVER SUPPORTS CLIENT-INITIATED INSECURE RENEGOTIATION
41	SERVER DOES NOT SUPPORT CLIENT-INITIATED INSECURE RENEGOTIATION
42	SERVER SUPPORTS OCSP STAPLING
43	SERVER DOES NOT SUPPORT OCSP STAPLING
44	SERVER-INITIATED SECURE RENEGOTIATION

45	SERVER SUPPORTS TLS COMPRESSION
46	SERVER DOES NOT SUPPORT TLS COMPRESSION
47	EC_POINT_FORMAT EXTENSION
48	HEARTBLEED
49	CVE-2014-0224
50	DROWN
51	POODLE OVER SSL
52	HTTPS SITE REDIRECTS TO HTTP
53	DNSCAA
54	ROBOT
55	SPF
56	DMARC
57	DKIM
58	EXPECT-CT
59	TLSv1.2 SUPPORTED
60	SERVER DOES NOT SUPPORT TLSv1.2
61	TLSv1.3 EARLY DATA
62	GOLDENDOODLE
63	Zombie POODLE
64	Sleeping POODLE
65	0-Length OpenSSL
206	CERTIFICATE CHAIN RELY ON EXPIRED CERTIFICATE
302	SERVER DOES NOT SUPPORT SERVER NAME INDICATION
304	SERVER DOES NOT SUPPORT EXTENDED MASTER SECRET
306	SERVER DOES NOT SUPPORT KEY SHARE
308	SERVER DOES NOT SUPPORT SUPPORTED VERSIONS
310	SERVER DOES NOT SUPPORT COOKIE

Appendix 6: List of Error messages

error_id	error

0	Unknown error. Please contact us.
1	You have performed [N] [ACTIONS] in the last 3 minutes. Please try again a bit later.
2	You have performed [N] [ACTIONS] in the last 24 hours. Buy premium API to run more tests.
3	Sorry, our systems are very busy now. Please try again in a few minutes.
4	You have running [N] concurrent [ACTIONS]. Please try again a bit later.
5	Sorry, there is a problem with your API key. Please double-check it or contact us.
6	Test is forbidden. Please contact us.
7	The domain name cannot be resolved. Please double-check it or contact us.
9	The domain name does not exist. Please double-check it or contact us.
10	An error has occured while checking DNS records of domain. Please double-check it or contact us.
11	Invalid IP address. Please double-check
12	Error with token. Our API has changed, please double-check it or contact us.
13	We could not conduct the requested test because a timeout occurred.
14	Arbitrary error from the engine.
16	Domain name was resolved in an invalid IP address.
17	An error occurred while encoding results.
18	Test does not exist.
19	PDF rendering problem has occurred.
20	Please register to [ACTION]
21	Your API key has exceeded the action-per-time limits. Please wait or contact us to increase the limits.
22	Your API key has expired. Please contact us to get a new one.
23	Your API key has been issued for another service.
24	Your API key does not exist.
25	Access denied for [IP].