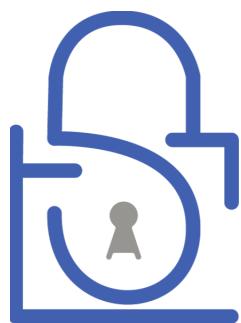


Target: localhost:4444

Analysis performed on 2025-10-29 17:46



TLSAssistant

Version 3.1.0

Confidentiality Disclaimer

This report contains confidential information intended solely for the use of the individual or entity to whom it is addressed. If you are not the intended recipient, please be advised that any disclosure, copying, distribution, or use of the contents of this report is strictly prohibited. If you have received this report in error, please notify the sender immediately and delete the original message.

Security Tool Report Disclaimer

The information provided in this report is the result of security testing conducted by [Your Company/Organization] using TLSAssistant. The purpose of this tool is to assess the security of TLS configurations. The findings and recommendations presented in this report are based on the specific conditions and configurations tested at the time of the assessment. We respectfully recommend verifying the identified vulnerability thoroughly.

No Warranty or Guarantee

This report is provided "as is," without any warranty, express or implied, concerning the accuracy, completeness, legal value or reliability of the information contained herein. [Your Company/Organization] disclaims all warranties, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

By reviewing this report, you acknowledge and agree to the terms and conditions outlined above.

Recap

Modules	localhost:4444
3SHAKE	Potentially Vulnerable
alpaca	Potentially Vulnerable
BEAST	Potentially Vulnerable
BREACH	Potentially Vulnerable
CSS Injection	Not Vulnerable
Missing Certificate Transparency	Not Vulnerable
CRIME	Not Vulnerable
DROWN	Potentially Vulnerable
Freak	Not Vulnerable
Heartbleed	Not Vulnerable
HSTS not preloaded	Potentially Vulnerable
HSTS not set	Potentially Vulnerable
HTTPS not enforced	Potentially Vulnerable
Logjam	Not Vulnerable
Lucky 13	Potentially Vulnerable
Bar Mitzvah	Not Vulnerable
RC4 NOMORE	Not Vulnerable
padding_oracle	Not Vulnerable
PFS	Not Vulnerable
Renegotiation attack	Not Vulnerable
ROBOT	Not Vulnerable
sslpoodle	Potentially Vulnerable
Sweet32	Potentially Vulnerable
Ticketbleed	Not Vulnerable
tlspoodle	Not Vulnerable

Detected vulnerabilities

localhost:4444

3SHAKE

CVE:Library dependent - CVSS3:Library dependent

Triple Handshake Attack

Due to the incorrect handling of the session identifier, an attacker is able to force two sessions to have the same Master Secret and ID. The attacker performs 3SHAKE by providing a server to which the victim deliberately connects. Once connected, the malicious server exploits the renegotiation mechanism to manipulate the session. The attack leads to a client impersonation that, by breaking both confidentiality and authentication, has a serious impact on the transmission.

Mitigation

Textual

The only acceptable mitigation is to use the [extended_master_secret](#) TLS extension. For this reason it is recommended to update the TLS library to a version that supports the aforementioned extension (e.g. OpenSSL v1.1.0+).

Apache

No snippet available

nginx

No snippet available

ALPACA

CVE:2021-3618 - CVSS3:7.4 (High)

Application Layer Protocol Confusion-Analyzing and Mitigating Cracks in TLS Authentication

ALPACA is an application layer protocol content confusion attack, exploiting TLS servers implementing different protocols but using compatible certificates, such as multi-domain or wildcard certificates. Attackers can redirect traffic from one subdomain to another, resulting in a valid TLS session. This breaks the authentication of TLS and cross-protocol attacks may be possible where the behavior of one protocol service may compromise the other at the application layer.

Mitigation

Textual

Enable the TLS Extensions SNI and ALPN.

Apache

Currently there is no snippet available to enable Strict ALPN.

To enable Strict SNI in Apache:

1. The first step is to create a dummy certificate for localhost with a 100-year lifespan:

```
sudo openssl genrsa -out /etc/ssl/private/localhost.key 2048
sudo openssl req -new -key /etc/ssl/private/localhost.key -subj /CN=localhost -out
/etc/ssl/certs/localhost.csr
echo subjectAltName=DNS:localhost | sudo openssl x509 -in
/etc/ssl/certs/localhost.csr -out /etc/ssl/certs/localhost.crt -req -signkey
/etc/ssl/private/localhost.key -days 36525 -extfile -
```

2. Create the virtual host configuration which uses that certificate so that TLS connections without a server name will now be directed to your dummy virtual host and fail:

```
<VirtualHost *:443>
ServerName localhost
SSLEngine On
SSLStrictSNIVHostCheck On
SSLCertificateKeyFile /etc/ssl/private/localhost.key
SSLCertificateFile /etc/ssl/certs/localhost.crt
<Location />
Require all denied
</Location>
</VirtualHost>
```

This must be placed in a location where it will be read before any other virtual host configuration, for example before all other virtual host configurations in:
`/etc/apache2/sites-available/example.com.conf`.

nginx

To enable Strict ALPN in nginx upgrade to version >=1.21.4.

To enable Strict SNI in nginx:

1. If you are using nginx>=1.19.4 edit your configuration file usually located in
`/etc/nginx/sites-enabled/default`

(if you changed your site conf name

`/etc/nginx/sites-enabled/YOURSITECONFIGURATION`); to look like this:

```
server {
listen 443 ssl default_server;
ssl_reject_handshake on;
}
```

```
server {
listen 443 ssl http2;
listen [::]:443 ssl http2;
```

```
server_name example.com;  
[...]  
}
```

2. If you are using nginx<1.19.4 follow [this guide](#)

BEAST

CVE:2011-3389 - CVSS2:4.3 (MEDIUM)

Browser Exploit and Against SSL TLS

The SSL protocol, as used in certain configurations in Microsoft Windows and Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and other products, encrypts data by using CBC mode with chained initialization vectors, which allows man-in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack (BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket API, (2) the Java URLConnection API, or (3) the Silverlight WebClient API, aka a "BEAST" attack.

Mitigation

Textual

Disable TLSv1.0 protocol support.

Apache

1. open your Apache configuration file (default: /etc/apache2/sites-available/default-ssl.conf);
2. search for the line starting with: SSLProtocol
- if it contains the substring **+TLSv1.0**, remove it;
- otherwise, add **-TLSv1.0** at the end of the line.

N.B. restart the server by typing: `sudo service apache2 restart`.

nginx

1. In a default situation, you can edit your website configuration /etc/nginx/sites-enabled/default
(if you changed your site conf name /etc/nginx/sites-enabled/YOURSITECONFIGURATION);
2. Inside `server {...}` brackets configuration, find `ssl_protocols`;
3. Remove `TLSv1.0` (if any). Make sure you have atleast another TLS protocol. If you can't find `ssl_protocols` you should be fine if your nginx is updated.

N.B. restart the server by typing: `sudo service nginx restart`.

BREACH

CVE:2013-3587 - CVSS3:5.9 (Medium)

Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext

By exploiting the information leakage provided by DEFLATE (compression algorithm), an attacker is able to retrieve the session cookie. In particular, the attacker guesses parts of the cookie, injects them in a valid client packet and analyzes the server's response. Thanks to the properties of a DEFLATE output, if the server's response is bigger than an untouched packet, then the guess is wrong.

Mitigation

Textual

Disable the HTTP compression mechanism.

Apache

- If Apache has been installed via package manager:
disable the DEFLATE module at OS level (e.g. on Ubuntu, run `sudo a2dismod deflate`);

- if Apache has been manually installed:

1. open your Apache configuration file (default: `/usr/local/apache2/conf/httpd.conf`);
2. search for the line containing : mod_deflate.so;
3. comment or delete it.

N.B. restart the server by typing: `sudo service apache2 restart`.

nginx

In a default situation, you can edit your website configuration
`/etc/nginx/sites-enabled/default`
(if you changed your site conf name
`/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);

- Do the following:

1. Edit this file with an editor;
2. add in your server configuration the string `gzip off;`.
If not missing, set it off by changing `gzip on;` to `gzip off;`

- Example:

For example, assuming this is your configuration:

```
server{  
    gzip on;  
}
```

Change it in:

```
server{
```

```
gzip off;  
}
```

If missing, just add `gzip off;` inside brackets.

N.B. restart the server by typing: `sudo service nginx restart`.

However this should be a fast temporary measure, use [this](#) instead.

Disabling gzip will disable compression in your website.

DROWN

CVE:2016-0800 - CVSS3:5.9 (Medium)

Decrypting RSA with Obsolete and Weakened eNcryption

By exploiting export-grade symmetric ciphers supported by SSLv2, an attacker is able to retrieve the MasterSecret that identifies the TLS session. The attack is performed by sniffing the network (or intentionally making requests to the server) looking for a vulnerable connection. Once found, the attacker repeatedly connects to the server using SSLv2 with export-grade RSA and a ClientMasterKey derived from the transmission he wants to decrypt. The MasterSecret can be used to decrypt the content of the transmission.

Mitigation

Textual

Disable SSLv2 protocol support.

Apache

1. open your Apache configuration file (default: `/etc/apache2/sites-available/default-ssl.conf`);
2. search for the line starting with: `SSLProtocol`
- if it contains the substring `+SSLv2`, remove it;
- otherwise, add `-SSLv2` at the end of the line.

N.B. restart the server by typing: `sudo service apache2 restart`.

nginx

1. In a default situation, you can edit your website configuration `/etc/nginx/sites-enabled/default` (if you changed your site conf name `/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);
2. Inside `server {...}` brackets configuration, find `ssl_protocols`;
3. Remove `SSLv2` (if any). Make sure you have atleast another TLS protocol. If you can't find `ssl_protocols` you should be fine if your nginx is updated.

N.B. restart the server by typing: `sudo service nginx restart`.

HSTS not preloaded

If the HSTS header is preloaded but the webserver does not have a valid certificate, the user will not be able to access the target website.

Mitigation

Textual

Check if your certificate has expired or it is not valid. If it is not valid, get a new certificate and deploy it on your server. If it has expired, renew it and deploy it on your server.

Apache

No snippet available

nginx

No snippet available

HSTS not set

Without the HSTS header, an attacker can use the SSL stripping attack to redirect all the HTTPS connection to their unsecure counterparts. By doing this, all the messages are sent in plaintext and can thus be manipulated.

Mitigation

Textual

Enable the HSTS header transmission within the webserver's settings.

Apache

1. open your Apache configuration file (default: `/etc/apache2/sites-available/default-ssl.conf`);
2. add the line `Header always set Strict-Transport-Security "max-age=31536000"`.

N.B. restart the server by typing: `sudo service apache2 restart` and be sure that `mod_headers` is enabled.

nginx

1. In a default situation, you can edit your website configuration `/etc/nginx/sites-enabled/default` (if you changed your site conf name `/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);
2. Add inside `server{...}` brackets: `add_header Strict-Transport-Security`

```
"max-age=31536000; includeSubdomains; preload";
```

N.B. restart the server by typing: `sudo service nginx restart`.

HTTPS not enforced

If HTTPS is not enforced, a client may be tricked into visiting the unsecure (HTTP) version of a website. This would allow an attacker to read and manipulate his messages.

Mitigation

Textual

For each HTTP connection the server must send a response containing:

1. a permanent redirect (i.e. [301 Moved Permanently](#));
2. a `Location` field indicating the proper URI to connect to (hostname preceded by `https://`).

Apache

1. open your Apache configuration file (default: `/etc/apache2/sites-available/default-ssl.conf`);
2. find the `VirtualHost` that handles the connections to port 80 (it starts with `<VirtualHost :80>`);
3. add the string `Redirect / https://website` where "website" is the URL you want to point the users to (e.g. `www.fbk.eu`).

N.B. restart the server by typing: `sudo service apache2 restart`.

nginx

1. In a default situation, you can edit your website configuration `/etc/nginx/sites-enabled/default` (if you changed your site conf name `/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);
2. add

```
server {  
    listen 80 default_server;  
    server_name _;  
    return 301 https://$host$request_uri;  
}
```
3. remove ALL other `listen 80` inside `server{...}` brackets (except for the previous one)

N.B. restart the server by typing: `sudo service nginx restart`.

Lucky 13

CVE:2013-0169 - CVSS2:2.6 (Low)

By exploiting the structure of the Cipher Block Chaining (CBC) mode, an attacker can infer the content of a transmission. The attack is performed by capturing, tampering (actually damaging) and re-transmitting the messages sent by the client to see how the server responds.

The attack, by breaching in the authentication mechanism, has a serious impact on the transmission.

Mitigation

Textual

Update the TLS library to a version that contains the custom mitigations (e.g. OpenSSL v1.0.1e+).

Apache

No snippet available

nginx

The best mitigation is to update the OpenSSL libraries. The fastest mitigation is to disable all CBC ciphers.

1. In a default situation, you can edit your website configuration
`/etc/nginx/sites-enabled/default`
(if you changed your site conf name
`/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);
2. Inside `server {...}` brackets configuration, find `ssl_ciphers`;
3. Remove any CBC-related cipher (even nested one).

N.B. restart the server by typing: `sudo service nginx restart`.

SSL POODLE

CVE:2014-3566 - CVSS3:3.4 (Low)

SSL Padding Oracle On Downgraded Legacy Encryption

By exploiting the missing validation of the padding bytes during decryption, an attacker is able to guess the session cookie. The attack is mounted by performing a MITM and requesting a SSLv3 connection between the client and the server. Once accepted, the attack is performed by modifying the padding in order to guess the cookie.

Mitigation

Textual

Disable SSLv3 protocol support.

Apache

1. open your Apache configuration file (default:
`/etc/apache2/sites-available/default-ssl.conf`);

2. search for the line starting with: SSLProtocol
- if it contains the substring **+SSLv3**, remove it;
- otherwise, add **-SSLv3** at the end of the line.

N.B. restart the server by typing: `sudo service apache2 restart`.

nginx

1. In a default situation, you can edit your website configuration `/etc/nginx/sites-enabled/default` (if you changed your site conf name `/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);
2. Inside `server {...}` brackets configuration, find `ssl_protocols`;
3. Remove `SSLv3` (if any). Make sure you have atleast another TLS protocol. If you can't find `ssl_protocols` you should be fine if your nginx is updated.

N.B. restart the server by typing: `sudo service nginx restart`.

Sweet32

CVE:2016-2183 - CVSS3:7.5 (High)

By exploiting the block size of the 3DES CBC ciphers, an attacker is able to mount a birthday attack. Once the victim has (forcefully) generated the needed amount of data (approximately 2^{32} blocks of data) its confidentiality can be affected due to the information leakage caused by the collisions in the cipher.

Mitigation

Textual

Disable support for 3DES cipher suites.

Apache

1. open your Apache configuration file (default: `/etc/apache2/sites-available/default-ssl.conf`);
2. find the line starting with: `SSLCipherSuite`;
3. add the string `:!3DES` at the end.

N.B. restart the server by typing: `sudo service apache2 restart`.

nginx

1. In a default situation, you can edit your website configuration `/etc/nginx/sites-enabled/default` (if you changed your site conf name `/etc/nginx/sites-enabled/YOURSITECONFIGURATION`);
2. Inside `server {...}` brackets configuration, find `ssl_ciphers`;
3. Remove 3DES (if any) and add `:!3DES` at the end.
4. Re run this tool. If it appears again, Remove IDEA (if any) and add `:!IDEA` at the end.

5. Re run this tool. If it appears again, Remove RSA (if any) and add :!RSA at the end.

N.B. restart the server by typing: `sudo service nginx restart`.